Statistical data analysis for biomarker discovery and Type 1 Diabetes prediction

Danmei Huang

Master thesis
UNIVERSITY OF HELSINKI
Department of Mathematics and Statistics

Helsinki, November 2, 2018

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Mathematics and Statistics

Tekijä — Författare — Author
Danmei Huang

Työn nimi — Arbetets titel — Title

Statistical data analysis for biomarker discovery and Type 1 Diabetes prediction

Oppiaine — Läroämne — Subject
Biostatistics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master thesis	November 2, 2018	43

Tiivistelmä — Referat — Abstract

Type 1 diabetes is a genetically related disease. The immune system attacks the pancreas so that no insulin can be secreted to regulate the blood glucose level. The cause of the disease is still unknown. To study Type 1 diabetes, researchers have collected time series microarray data for thousands of genes from individuals divided into case and control groups. We aim to detect genes that show significant differences between cases and controls by analyzing the data. These genes may be used as biomarkers for Type 1 diabetes prediction in the future.

We present 4 statistical methods for analyzing this Type 1 diabetes gene expression data, based on different considerations. We provide detailed introductions to the methods that are used in the analysis of the thesis. In particular, we show that Gaussian process regression is actually an extension of linear regression.

The first method, standard linear regression, assumes both cases and controls follow the same linear model, except that the cases exhibit large variation at some time point. Those time points with large variation are also known as outliers. We can estimate their predictive distribution and calculate their p-values to check the significance.

The second method, Bayesian linear regression, considers the variation of the point estimates (maximum likelihood) in the standard linear regression. We place priors on the parameters such that the uncertainty of the parameters can be integrated out. The estimates are generally more robust than the standard linear regression.

The third method, Gaussian process regression, assumes both cases and controls follow the same non-linear model. This is in contrast to the linear model in the previous two methods. Gaussian process is a non-parametric model that is very flexible. The squared exponential kernel used in this thesis is able to model almost all smooth functions. After the fitting of the data, we can calculate the predictive distribution of data points of the cases. Then we can detect the outliers by checking their p-values.

The fourth method, Gaussian process model comparison, models the difference between cases and controls as a whole. Cases may be systematically different to controls, or not. We use a shared model to model them jointly and an independent model to model them separately. After that we calculate the Bayes factor between the two models. If cases and controls are very similar, they will follow the shared model with a higher marginal likelihood. If they differ a lot, the independent model is preferred.

We apply the above four methods to the microarray data, which contains 49386 genes for 6 case-control pairs. We find 4956, 661 and 2797 significant genes using the first three methods with Bonferroni corrections to the p-values. The numbers are 43276, 3584 and 25149 if we use Benjamini-Hochberg correction. The fourth method suggests 722 significant genes with the log Bayesian factor less than -5.

We presents some example significant genes that show difference between cases and controls. They clearly show the expected difference between cases and controls. The example results suggest in general Gaussian process models fit the data better than linear regression models.

The top hits (genes) provided by the methods remain to be validated by more biological experiments.

Avainsanat — Nyckelord — Keywords
Type 1 diabetes, T1D, Gaussian process, outlier

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Biological background

Diabetes is a disease associated with high blood sugar concentration. The blood sugar (or glucose) is regulated by a hormone called *insulin* and a hormone called *glucagon*, as shown in Figure 1. When the blood sugar concentration increases, insulin will be secreted from beta cells in the pancreas. Insulin helps to move the blood sugar into the cells of our body, such that the cells can utilize the sugar to generate energy for mainteining their metabolism. After the secretion of insulin, the blood sugar level decreases to a normal level and the beta cells in pancreas will stop secreting insulin. When the blood sugar level becomes low, alpha cells in the pancreas will secrete glucagon into the blood. Stimulated by glucagon, the liver will release glucose by decomposing glycogen stored in the liver to maintain a suitable blood sugar level. In a word, the blood sugar level is regulated by the pancreas secreting insulin and glucagon alternatively.

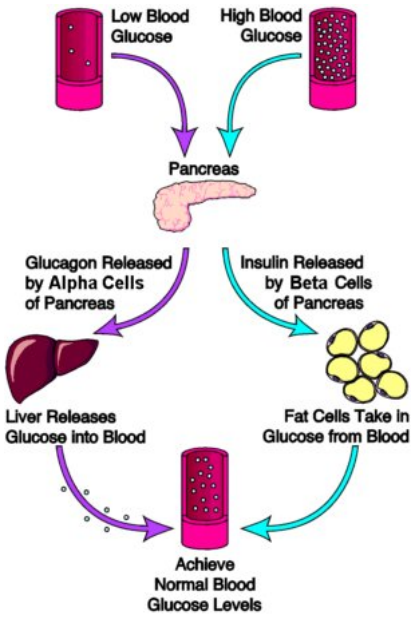

Figure 1: Blood sugar regulation. Glucagon increases blood sugar level and insulin decreases it. Reprinted from [13]

Since insulin plays an important role in regulating the blood sugar level, diabetes is a natural consequence when insulin does not function as expected. When no insulin is available or the body is resistant to insulin, the blood sugar can not move into tissues and the sugar starts to accumulate in the blood. The body will try to get rid of the extra sugar in the blood by urination. This

leads a symptom of frequent urination. Lots of water is lost in the process of frequent urination, which is the reason why diabetes patients feel very thirsty. Since tissues cannot get enough blood sugar, diabetes patients often feels very hungry and tired. There are also other symptoms of diabetes patients such as blurred vision, nausea etc, caused by high blood sugar level via more complex mechanisms. We will not further discuss these mechanisms.

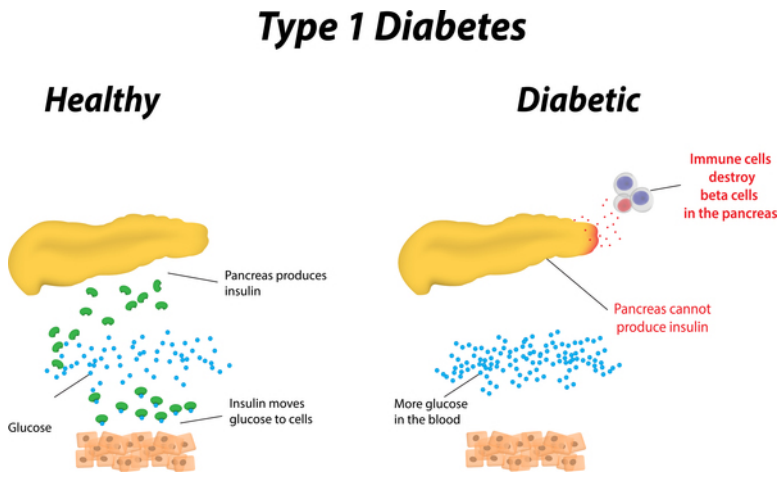

Figure 2: Type 1 diabetes. Since beta cells in pancreas are killed by the immune system, no insulin is secreted and the glucose starts to accumulate in the blood. Reprinted from [10]

There are three main types of diabetes: type 1 diabetes, type 2 diabetes and gestational diabetes. Type 1 diabetes is related with the dysfunction of beta cells in the pancreas such that no insulin is produced in diabetes patients. In type 2 diabetes patients, insulin is still produced but the body is resistant to insulin such that insulin cannot decrease the blood sugar level anymore. In other words, blood sugar are not transported into tissues as usual even if insulin level increases. Gestational diabetes is also related with insulin resistance but only occurs in pregnant women. This thesis focuses on analyzing type 1 diabetes data and thus we provide a more detailed explanation about type 1 diabetes.

Type 1 diabetes (T1D) is a genetic disease and usually occurs at a young age. In the body of type 1 diabetes patients, the immune system attacks the beta cells in the pancreas due to genetic factors and environmental factors such as virus infection [1, 6]. In other words, the immune system cannot recognize the beta cells as normal tissues and kills them. As beta cells being killed, insufficient or no insulin is produced to decrease the blood sugar level. Then type 1diabetes starts. The overview of T1D is shown in Figure 2.

Genetic mutations [17] related with immune system, together with environmental factors such as virus infection [6], can lead to disorders in the immune system. When the immune system starts to attack the beta cells, it always comes along with autoantibodies in the blood. An autoantibody [5] is an antibody produced by the immune system to target one's own proteins such that they lose their normal functions. If any autoantibody is detected in the blood, we call it **sero conversion**. Sero conversion only says that the immune system is attacking one's own proteins, but it may not be beta cells and T1D is not necessarily going to occur in the future. As shown in Figure 3, T1D patients must experience sero conversion at an earlier time, but sero conversion does not necessarily lead to T1D at a later stage. Currently the cause of T1D is unknown and no treatments are available to prevent its occurrence. Fortunately, T1D patients can maintain life quality by insulin injection and diet control.

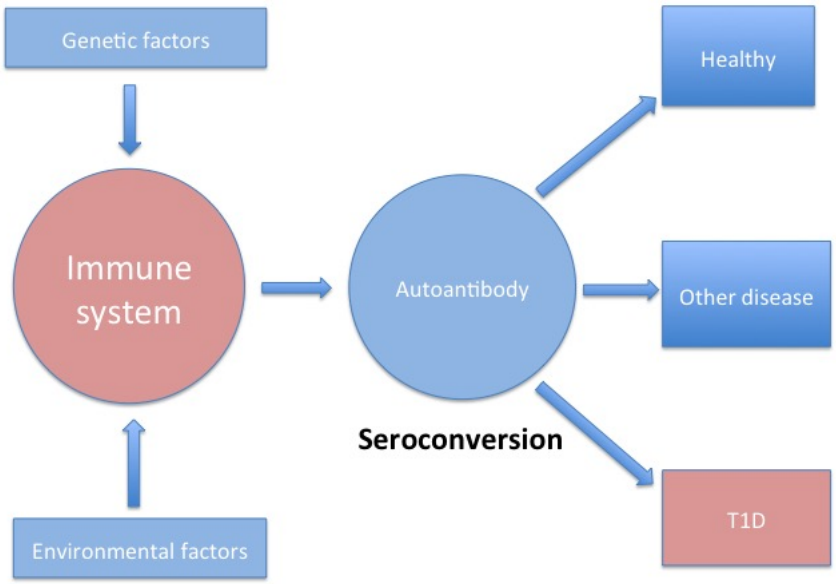

Figure 3: Sero conversion.

In order to study the cause of T1D, a longitudinal case-control study (Figure 4) is usually adopted, where genes/proteins are measured at similar time points for all individuals. Cases refers to T1D patients and controls refers to healthy individuals with similar genetic and clinic background, such as human leukocyte antigen (HLA) allele gender, age etc. Cases and controls are

matched so that we can maximumly exclude the effects of irrelevant factors. In this thesis, we hypothesis that the relevant factors of T1D are genes with different expression levels or patterns between cases and controls. Microarray [16] is used to measure the gene expression levels in the blood of cases and controls. Microarray is a technique that measures the gene expression levels for hundreds and thousands of genes at the same time. We aim to find genes that either show an expression level difference between cases and controls.

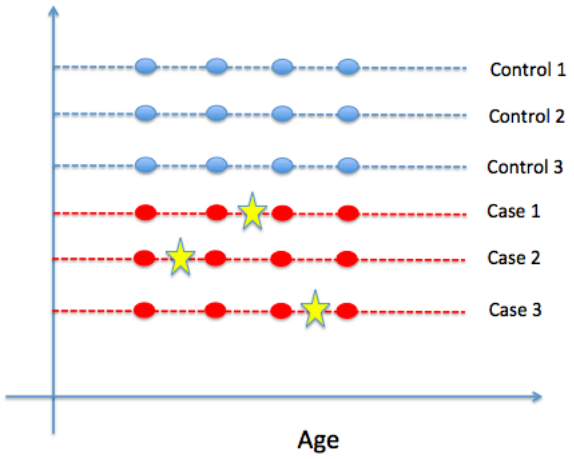

Figure 4: Case-control longitudinal study. Cases are colored in red and controls are colored in blue. The x-axis is age and the dots are the time points blood samples are collected. The stars are the sero conversion time points.

2 Introduction

In this section, we provide some statistical backgrounds for the methods to be used.

2.1 Frequentist & Bayesian statistics

Random variable (RV or r.v.) is the key object in statistics. A random variable can be discrete or continuous. A random variable is a function of the outcome of some random event, which measures the probability of a certain outcome. A discrete random variable have a probability mass function (pmf) and a continuous random variable have a probability density function (pdf). The value of a probability mass function must be between 0 and 1. The value of a probability density function can be greater than 1 at certain point, as long as the integral over its domain is 1.

There are two types of statistics, Frequentist statistics and Bayesian statistics. Frequentist statistics is more traditional and heavily used in linear regression. The central idea is to assume a unknown true value for each parameter. We can only approximate the true value with more data, but we will never get the true value unless we have infinite amount of data. Bayesian statistics admits the uncertainty and believes all values are possible and treat the parameter as a random variable. It then develops a principled way to quantify the uncertainty using the Bayes' theorem. In short, we deal with point estimate more in frequentist statistics and distribution more in Bayesian statistics.

In Bayesian statistics, we are usually interested in three things: *prior*, *likelihood* and *posterior*, which are the three key components in Bayes' Theorem. Let us say we are interested in a parameter θ, which is used to generate the observed data x. The *prior* $p(\theta)$ refers to the distribution of the parameter θ without seeing any data. It represents a subjective understanding of the parameter, which is usually very vague in most cases to allow similar probabilities for different values. The *likelihood* $p(x|\theta)$ is a function for calculating the probability of generating the data given the parameter θ. As the value of θ changes, the likelihood changes accordingly. The *posterior* $p(\theta|x)$ refers to the distribution of the parameter after seeing the data, which represents an updated understanding of the parameter given the data. The Bayes' Theorem is thus given as

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}, \tag{1}$$

which follows from the product rule in probability theory [3]. Note that the

marginal probability $p(x)$ is given by

$$p(x) = \int_{\theta} p(x|\theta)p(\theta)d\theta, \tag{2}$$

by following the sum rule in probability theory [3].

2.2 Model inference

As we can see, Bayes' theorem provides a principled way to model the uncertainty of the parameter. In practice, the posterior of the parameter is usually more peaked than the prior, which is a sign of reduced uncertainty. Deriving the posterior, however, is not necessarily easy depending on the calculation of the marginal probability (Eq. (2)). If we can get an analytic solution of Eq. (2), then it is much easy to derive the posterior distribution. If analytic solution is not available, we need to resort to Markov chain Monte Carlo (MCMC). The process of inferring the posterior distribution given the input data is called *model inference*.

2.2.1 Point estimates

Sometimes inferring the posterior distribution is computationally unfeasible or time consuming, we can get a point estimate for the parameter and estimate the confidence intervals. This is more commonly used in frequentist statistics and less popular in Bayesian statistics. Two types point estimates are generally used: maximum likelihood estimate (ML) and maximum a posterior estimate (MAP). ML estimate refers to the parameter value that maximizes likelihood:

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(x|\theta). \tag{3}$$

MAP estimate refers to the parameter value that maximizes that posterior probability:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta|x) = \arg\max_{\theta} p(x|\theta)p(\theta). \tag{4}$$

Note that the normalizing constant $p(x)$ is a constant with respect to the parameter θ, therefore we only need to maximize the numerator of the Bayes' formula (Eq. (1)).

After the point estimate is obtained, we are usually interested in deriving a confidence interval for the parameter. The confidence interval mean the true parameter value falls in this interval with a certain probability. A common treatment is to assume a Gaussian distribution for the parameter. The ML

or MAP estimate $\hat{\theta}$ is the mean and the variance σ^2 is estimated separately depending on the specific problem. Then we can use the standard conclusions of Gaussian distribution to derive the confidence interval given a confidence value, e.g. $(\hat{\theta} - 2\sigma, \hat{\theta} + 2\sigma)$ corresponds to the 95% confidence interval, as shown in Figure 5. If σ^2 is large, we are less certain of the true parameter value since the interval is large. On the opposite, we are quite confident of the true parameter value when σ^2 is small.

2.2.2 p-value

p-value [8] is widely used in frequentist statistics to check the significance of a *null hypothesis*. A null hypothesis represent the belief how a parameter should be distributed, which is usually in the form of a Gaussian distribution. For a given parameter $\theta \sim N(\mu, \sigma^2)$ according to the null hypothesis, its p-value is given by $P(x \geq \theta)$ (when $\theta > \mu$) or $P(x \leq \theta)$ ($\theta <= \mu$). It describes how likely an at least as extreme outcome x is generated under the null hypothesis, i.e. the outcome x is greater (or lower) than or equal to the given parameter θ. The smaller p-value is, the lower probability to observe a value greater (or lower) than the given parameter θ, thus hints for a higher significance of an alternative hypothesis. 5% is usually used as the threshold for p-value, which corresponds to the integrated probability of the region $[\mu + 2\sigma, +\infty)$ (or $(-\infty, \mu - 2\sigma]$) in Figure 5.

2.2.3 False discovery rate (FDR)

False discovery rate arises when we need to perform statistical testing for many times. For example, if we need to perform the same testing for 10,000 genes and the significance threshold is set to 5%, it is quite likely that we see many genes that are significant, i.e. reject the null hypothesis. However, many of these significant genes are not significant, they should be generated by the null hypothesis simply by chance. This kind of error is called *type 1 error*. The rate of type 1 error is called false discovery rate. In order to reduce the false discovery rate, we can use the Holm–Bonferroni method [11] or the Benjamini and Hochberg method [2], as shown in Algorithm 1 and 2, respectively. The Holm–Bonferroni method is more strict than the Benjamini and Hochberg method, i.e. it usually leads to less number of significant hits

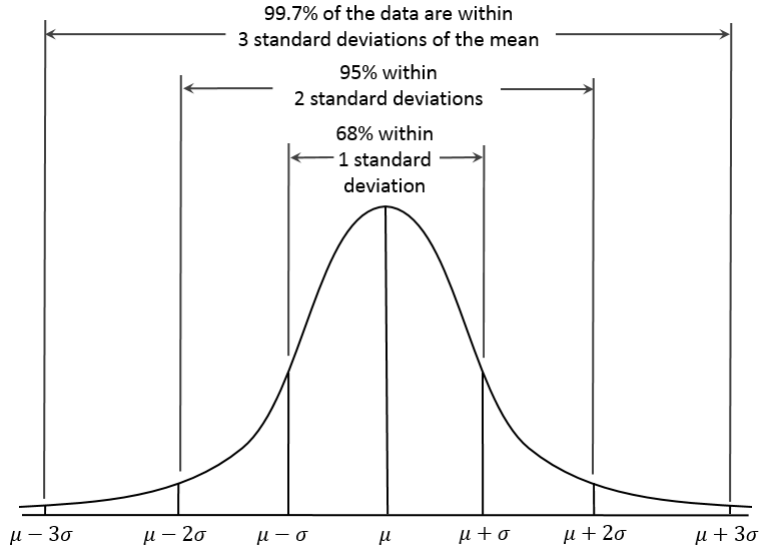

Figure 5: Empirical confidence intervals of Gaussian distribution. The confidence intervals can be calculate for any given confidence value. However, confidence intervals that are multipliers of standard deviation are widely used empirically. Reprinted from [12].

after correction.

Algorithm 1: The Holm–Bonferroni method

Input : Null hypothesis $H_1, \cdots, H_m$ and the corresponding p-values $P_1, \cdots, P_m$. Significance level α

Sort the p-values in ascending order $P_{(1)}, \cdots, P_{(m)}$, which corresponds to $H_{(1)}, \cdots, H_{(m)}$

Find $k = \arg\min_k P_{(k)} > \frac{\alpha}{m+1-k}$

Reject $H_{(1)}, \cdots, H_{(k-1)}$ and accept $H_{(k)}, \cdots, H_{(m)}$

If $k = 1$, accept all null hypothesis. If k do not exist, reject all null hypothesis.

Output : A list of accepted/rejected null hypothesis.

Algorithm 2: The Benjamini-Hochberg method

Input : Null hypothesis $H_1, \cdots, H_m$ and the corresponding p-values $P_1, \cdots, P_m$. Significance level α

Sort the p-values in ascending order $P_{(1)}, \cdots, P_{(m)}$, which corresponds to $H_{(1)}, \cdots, H_{(m)}$

Find $k = \arg\max_k P_{(k)} \leq \frac{k}{m}\alpha$

Reject $H_{(i)}$ for all $i = 1, \ldots, k$

Output : A list of accepted/rejected null hypothesis.

2.2.4 Conjugate prior

Since fast inference of the posterior distribution is very important, researchers try to use prior and likelihood pairs which can lead to analytic solutions of Eq. (2) when designing the model. Such priors are called *conjugate priors*, which means the posterior distribution is in the same family of the prior, i.e. the prior and posterior share the same function form by with different parameters. Here we take Beta-Bernoulli conjugate pair as an example. The prior $p(\theta)$ is

$$beta(\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}, \tag{5}$$

and the Bernoulli likelihood $p(x|\theta)$ is

$$p(x|\theta) = \theta^x(1-\theta)^{1-x}, \tag{6}$$

where $x \in \{0, 1\}$. The joint probability of θ and x is thus

$$\begin{aligned} p(x,\theta) &= p(\theta)p(x|\theta) \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}\theta^x(1-\theta)^{1-x} \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{(\alpha+x)-1}(1-\theta)^{(\beta+1-x)-1} \\ &= \left(\frac{\Gamma(\alpha'+\beta')}{\Gamma(\alpha')\Gamma(\beta')}\theta^{\alpha'-1}(1-\theta)^{\beta'-1}\right)\cdot\left(\frac{\Gamma(\alpha')\Gamma(\beta')}{\Gamma(\alpha'+\beta')}\cdot\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right), \end{aligned} \tag{7}$$

where $\alpha' = \alpha + x$ and $\beta' = \beta + 1 - x$. Note that the front part of Eq. (7) is the pdf of $beta(\alpha', \beta')$ and the end part is a constant with respect to θ, which means the front part equals to 1 if integrated over θ and thus the analytic solution for Eq. (2) equals the second part, i.e.

$$p(x) = \int_\theta p(x|\theta)p(\theta)d\theta = \frac{\Gamma(\alpha')\Gamma(\beta')}{\Gamma(\alpha'+\beta')}\cdot\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{8}$$

Inserting Eq. (7) and (8) into the Bayes' formula Eq. (1), we can derive the posterior distribution is

$$p(\theta|x) = \frac{\Gamma(\alpha'+\beta')}{\Gamma(\alpha')\Gamma(\beta')}\theta^{\alpha'-1}(1-\theta)^{\beta'-1}, \tag{9}$$

which is actually $beta(\alpha', \beta')$. We observe that both the prior and posterior are in the form of beta distribution, where the only difference lies in the hyperparameters α and β. This means we only need to update the hyperparameters to conduct the inference, even without the need to calculate the

marginal constant Eq. (8). Therefore it is very fast and that is the advantage of conjugate priors.

Another commonly used conjugate prior for multivariate Gaussian with unknown mean and variance is normal-inverse-Wishart distribution [7]. Let us denote the prior by

$$(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim NIW(\boldsymbol{\mu}_0, \kappa_0, \nu_0, \Psi). \tag{10}$$

The corresponding posterior after observing $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ are given by

$$(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim NIW(\boldsymbol{\mu}', \kappa', \nu', \Psi') \tag{11}$$

where

$$\boldsymbol{\mu}' = \frac{\kappa_0 \boldsymbol{\mu}_0 + n\bar{\boldsymbol{x}}}{\kappa_0 + n} \tag{12}$$

$$\kappa' = \kappa_0 + n \tag{13}$$

$$\nu' = \nu_0 + n \tag{14}$$

$$\Psi' = \Psi + C + \frac{\kappa_0 n}{\kappa_0 + n}(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_0)(\bar{\boldsymbol{x}} - \boldsymbol{\mu}_0)^T \tag{15}$$

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \tag{16}$$

$$C = \sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T. \tag{17}$$

This distribution is rather complex, we do not provide its density functions and explain the parameters here. But it is easy to calculate the pdf and sample from this distribution once the parameters are given.

2.2.5 Markov chain Monte Carlo (MCMC)

Sometimes we are not able to find conjugate priors for our problem, then we need to resort to Markov chain Monte Carlo (MCMC). In this case we are facing two difficulties in deriving the posterior 1) unknown or difficult to compute normalizing constant $p(x)$ and 2) the joint distribution $p(x, \theta)$ is in a function form that does not lie in existing well studied probability distributions. These two coupled difficulties make it difficult to study the actual posterior distribution directly, i.e. the shape of pdf is not known. To circumvent this problem, we can draw samples from the posterior distribution using MCMC. The whole set of posterior samples is then used to approximate the actual posterior distribution.

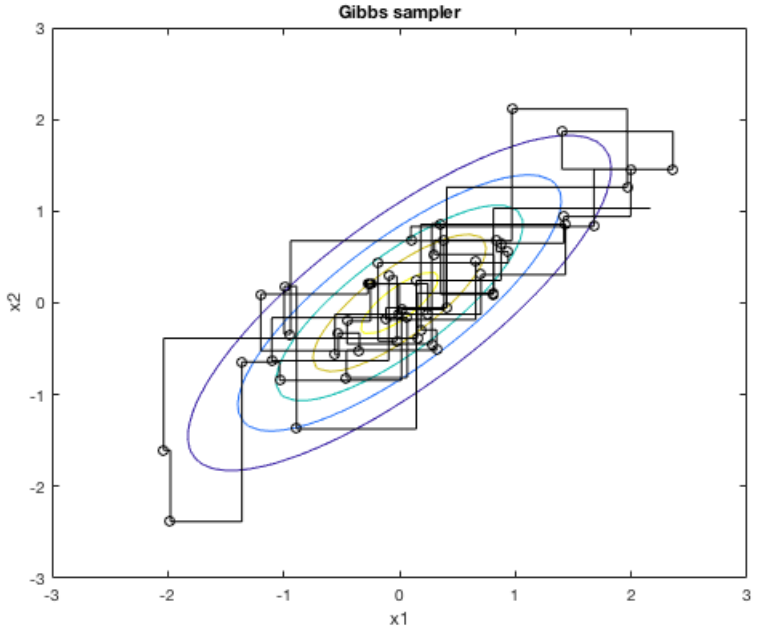

Figure 6: First 50 samples of Gibbs sampler from a bivariate normal distribution with μ=[0 0] and Σ=[1 0.8; 0.8 1]. It can be seen that Gibbs sampler samples from the conditional distribution alternatively, thus the samples are correlated.

There are two general types of MCMC techniques. One is called Gibbs sampler and the other is called Metropolis–Hastings sampler. Both samplers draw correlated samples for N steps, where N should be reasonably large to ensure we get enough effective samples. The central idea is to make the posterior distribution to be the stationary distribution of a Markov chain. Given the detailed balance condition [9], it is proved that we will reach the stationary distribution of Markov chain after sampling infinite amount of samples [4], no matter which initial value we start from.

Algorithm 3 shows the procedures of Gibbs sampler. Gibbs sampler is suitable for joint pdf that can be written in a product form of independent known pdfs. Then we can keep all other parameters fixed and calculate the conditional distribution of the left out parameter, which is in a form of known distributions. Figure 6 shows an example of sampling from a 2-dimensional multivariate normal distribution using Gibbs sampler. The sampling path can be seen from the horizontal and vertical bars in the figure.

Algorithm 3: Gibbs sampler algorithm

Result: Correlated parameter values $(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(T)})$
Set $t = 0$
Initialize the parameter $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \cdots \theta_D^{(0)})$
for $t \leftarrow 1$ **to** T **do**
  for $i \leftarrow 1$ **to** D **do**
    draw $\theta_i^{(t)}$ from $p(\theta_i | \boldsymbol{\theta}_{-i}^{(t-1)})$, where $\boldsymbol{\theta}_{-i}^{(t-1)} = \boldsymbol{\theta}^{(t-1)} \setminus \theta_i^{(t-1)}$
  end
end

Algorithm 4 shows the steps for Metropolis-Hastings sampler. This sampler requires a proposal distribution from which we can sample parameter values. If the proposal distribution's shape is similar to our posterior distribution, then the sampler will work with maximum efficacy. If the proposal distribution is the same as the target distribution, we can see the acceptance ratio r is 1, which means every proposed sample will be accepted. Note that we are able to calculate the ratio of pdf between two parameter values since the the normalization constant cancels out. Figure 7 shows an example of using the Metropolis-Hastings sampler to sample from the sample multivariate normal

distribution in Figure 6.

Algorithm 4: Metropolis-Hasting sampler algorithm

Result: Correlated parameter values $(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(T)})$
Set $t = 0$
Initialize the parameter $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \cdots \theta_D^{(0)})$
for $t \leftarrow 1$ **to** T **do**
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(t-1)}$
  Sample $\boldsymbol{\theta}'$ from the proposal distribution $g(\boldsymbol{\theta}'|\boldsymbol{\theta})$
  Calculate the acceptance ratio $r = min(1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})} \frac{g(\boldsymbol{\theta}|\boldsymbol{\theta}')}{g(\boldsymbol{\theta}'|\boldsymbol{\theta})})$, where $\pi(\boldsymbol{\theta})$ denotes the target distribution.
  Draw a random number $u \sim Uni(0, 1)$
  if $u \leq r$ **then**
    Accept, $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}'$
  else
    Reject, $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}$
  end
end

After MCMC inference, we will get a list of correlated parameter values (or samples) $(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(T)})$ as the result. The result cannot be used directly since 1) the samples are highly correlated and 2) the convergence of the Markov chain is not checked. To make the samples less correlated, *burn-in* and *thinning* are widely used [4]. Burn-in refers to removing a certain amount of sample from the beginning of the Markov chain. Thinning refers to extract one out of every k consecutive samples. After burn-in and thinning, we expect the extracted samples to be randomly distributed according to the posterior distribution, which is usually termed as *mixing* well. In other words, if we take the same amount of consecutive samples from two different locations of the chain, they should exhibit similar properties.

Following the idea above, potential scale reduction factor (PSRF) $\hat{R}$ [7] is proposed to check the convergence. The central idea is to split the whole Markov chain into m equal fragments and check the variances within and between the fragments. Let us assume each fragment has n samples and we denote the ith fragment of a specific parameter by $(\theta_{i1}, \theta_{i2}, \cdots, \theta_{ij}, \cdots, \theta_{in})$. The average within fragment variance is

$$W = \frac{1}{m} \sum_{i=1}^{m} \sigma_i^2 \tag{18}$$

where $\sigma_i^2 = \frac{1}{n-1} \sum_{j=1}^{n} (\theta_{ij} - \bar{\theta_i})$ is the within fragment variance and $\bar{\theta_i} = \frac{1}{n} \sum_{j=1}^{n} \theta_{ij}$ is the average parameter value of the ith fragment. The between

fragment variance is

$$B = \frac{n}{m-1}\sum_{i=1}^{m}(\bar{\theta}_i - \bar{\bar{\theta}}), \tag{19}$$

where $\bar{\bar{\theta}} = \frac{1}{m}\sum_{i=1}^{m}\bar{\theta}_i$ is the average parameter over all fragments. The potential scale reduction factor $\hat{R}$ is then given as

$$\hat{R} = \sqrt{\frac{\sigma_\theta^2}{W}}, \tag{20}$$

where $\sigma_\theta^2 = (1-\frac{1}{n})W + \frac{1}{n}B$ is the harmonized variance. $\hat{R}$ should be close to 1 if the Markov chain has converged. As suggested in [7], we require $\hat{R} \leq 1.1$ in practice. If $\hat{R} > 1.1$, it means the chain has not converged and we need more samples.

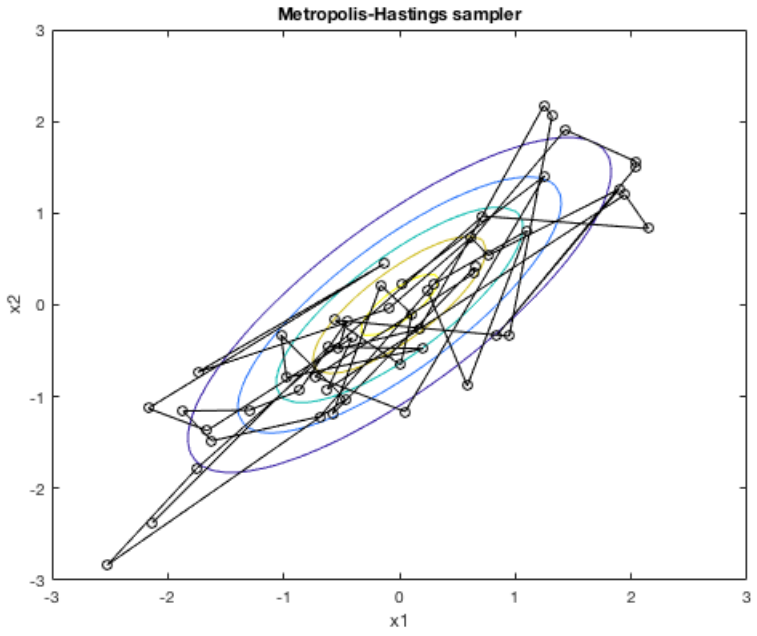

Figure 7: 50 samples of MH sampler after burn-in and thinning from a bivariate normal distribution with μ=[0 0] and Σ=[1 0.8; 0.8 1]. It can be seen that the samples are less correlated than that in Figure 6

With converged MCMC samples $(\theta_1, \theta_2, \cdots, \theta_i, \cdots, \theta_n)$, we are able to do various analysis of the parameter to be inferred. We could derive the empirical distribution by making a histogram from the samples. We could

also use the samples to calculate the expectation of some function $h(y, \theta)$ we are interested in, as shown in the following equation.

$$E_{\theta|x}(h(y, \theta|x)) = \int_{\theta} h(y, \theta) p(\theta|x) d\theta = \frac{1}{n} \sum_{i=1}^{n} h(y, \theta_i), \tag{21}$$

where $p(\theta|x)$ is the posterior distribution of θ. If y is new data, then Eq. (21) is the predictive probability, i.e. how likely y is generated by considering all configurations of θ.

2.3 Model selection

In Bayesian statistics, models generally have hierarchical components. This is because all parameters are treated as distributions, i.e. the hyper-parameters are also modeled as distribution. If we keep on imposing hyer-parameters over hyper-parameters, we can easily build up complex hierarchical models. Also, there exists lots of dependencies in real world applications, which is another reason for the popularity of hierarchical models. These models are generally called graphical models. Figure 8 shows an example of hierarchical model taken from

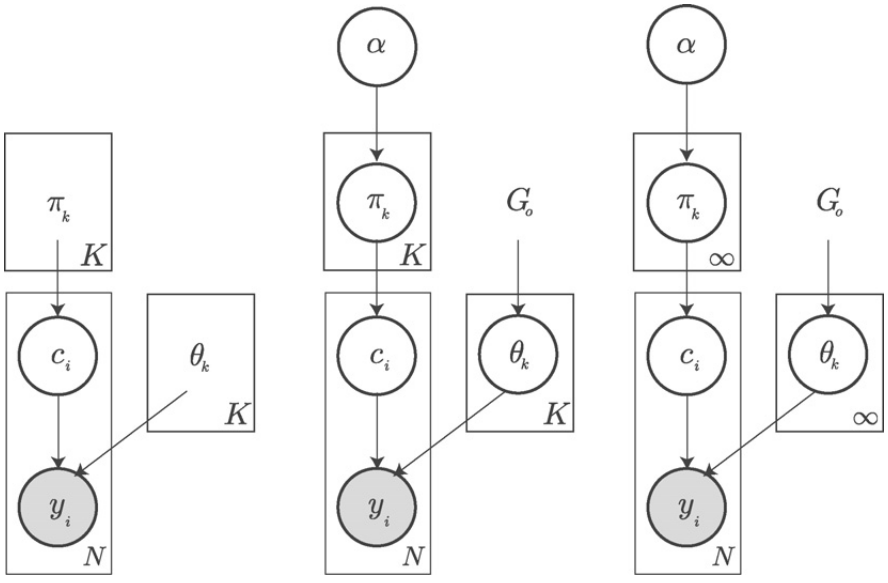

Figure 8: An example of graphical model. Circles are random variables. Rectangles mean repetitions according to the number in its lower right corner. Shaded circles are observations.

For the same data, we can design different graphical models by adopting different hypothesis. For example, we can assume all data points are generated

by a single Gaussian distribution, or assume they are generated by a mixture of Gaussian distribution. The next natural question is then which model is better. Direct comparison of the two models is not possible since they have different structures and thus also parameters. We resort to *marginal likelihood* to compare two different models, shown as follows.

$$p(x|M) = \int_{\Theta} p(x, \boldsymbol{\theta}|M) d\boldsymbol{\theta}, \tag{22}$$

where x is the observation, M is the model and $\boldsymbol{\theta}$ are the parameters. As can be seen, the marginal likelihood integrate out all intermediate parameters, which allows the comparison of the two models. Integration of the parameters in Eq. (22) can be derived using different methods: conjugate priors, MCMC, Laplace approximation [15]. When we use Laplace approximation to calculate Eq. (22), it can be shown that the approximation of Eq. (22) is equivalent to *Bayesian information criterion* (BIC).

To compare two models M_1 and M_2, we use *Bayes factors* defined as follows.

$$BF = \frac{p(x|M_1)}{p(x|M_2)} \tag{23}$$

When using Bayes factors, we implicitly assign equal prior probability to both models. If $BF \geq 1$, it means M_1 is more favorable than M_2; otherwise M_2 is more favorable. In practice, we require $BF > 5$ for M_1 to be selected.

To compare multiple models, we follow the same idea by getting $p(M_i)$ and $p(x|M_i)$ for each model i, then use the Bayes' formula Eq. (1) to derive posterior distribution of each model.

$$p(M_i|x) = \frac{p(x|M_i)p(M_i)}{\sum_j p(x|M_j)p(M_j)} \tag{24}$$

2.4 Gaussian process

The method for finding differences between cases and controls is based on Gaussian process. Here we provide an explanation to Gaussian process starting from linear regression. We will use notations from [3] for this section. Since the derivations are complicated, we will provide the relevant main conclusions directly.

2.4.1 Linear regression

We assume the target variable $\boldsymbol{t}$ is a N-dimensional column vector $\boldsymbol{t} = (t_1, t_2, \cdots, t_N)^T$. The input data is denoted by $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)^T$, where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iM})^T$ represents the M-dimensional input variables vector

for the ith target variable. In linear regression, we assume each target variable is the sum of a linear function $y(\boldsymbol{x}, \boldsymbol{w})$ and a noise ϵ, i.e.

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon, \tag{25}$$

where $\boldsymbol{w} = (w_1, w_2, \cdots, w_M)^T$ is simply a linear combination of the input variables.

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=1}^{M} w_j x_j = \boldsymbol{w}^T \boldsymbol{x} \tag{26}$$

The noise is usually chosen to be zero mean Gaussian $\epsilon \sim N(0, \beta^{-1})$, where β is the precision (inverse variance) of the Gaussian distribution. Thus the likelihood for a single target variable is also Gaussian.

$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = N(t|y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}) = N(t|\boldsymbol{w}^T \boldsymbol{x}, \beta^{-1}) \tag{27}$$

The likelihood for all target variables is thus a multivariate Gaussian

$$p(\boldsymbol{t}|X, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} N(t_n|\boldsymbol{w}^T \boldsymbol{x}_n, \beta^{-1}) = N(\boldsymbol{t}|X\boldsymbol{w}, \beta^{-1}\boldsymbol{I}), \tag{28}$$

where $\boldsymbol{I}$ is the identity matrix whose elements in the diagonal all equal to 1.

Let us now assume the noise precision β is given (fixed), then the random variables of interest is the linear coefficient vector $\boldsymbol{w}$. In this setting, we will need to specify a prior for $\boldsymbol{w}$ and figure out its posterior $p(\boldsymbol{w}|\boldsymbol{t}, X, \beta)$ after fitting the data. Here we choose the following multivariate Gaussian distribution as the prior, which a conjugate prior of the Gaussian likelihood.

$$p(\boldsymbol{w}|\alpha) = N(\boldsymbol{w}|\boldsymbol{0}, \alpha^{-1}\boldsymbol{I}) \tag{29}$$

Due to the conjugacy, the posterior distribution of $\boldsymbol{w}$ is also a multivariate Gaussian.

$$p(\boldsymbol{w}|\boldsymbol{t}, X, \alpha, \beta) = N(\boldsymbol{w}|\boldsymbol{m}_N, \boldsymbol{S}_N), \tag{30}$$

where

$$\boldsymbol{m}_N = \beta \boldsymbol{S}_N X^T \boldsymbol{t} \tag{31}$$

$$\boldsymbol{S}_N^{-1} = \alpha \boldsymbol{I} + \beta X^T X \tag{32}$$

Here we provide an intuitive example of Bayesian linear regression using the above conclusions, as shown in Figure 9. We generate the data using a linear model $t = -0.3 + 0.5x + \epsilon$, where the i.i.d noise $\epsilon \sim N(0, 0.2^2)$. According to our model specification, the linear coefficients are $\boldsymbol{w} = (-0.3, 0.5)$ and the input variables are $\boldsymbol{x} = (1, x)$. We use a vague Gaussian prior $N(\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$ for

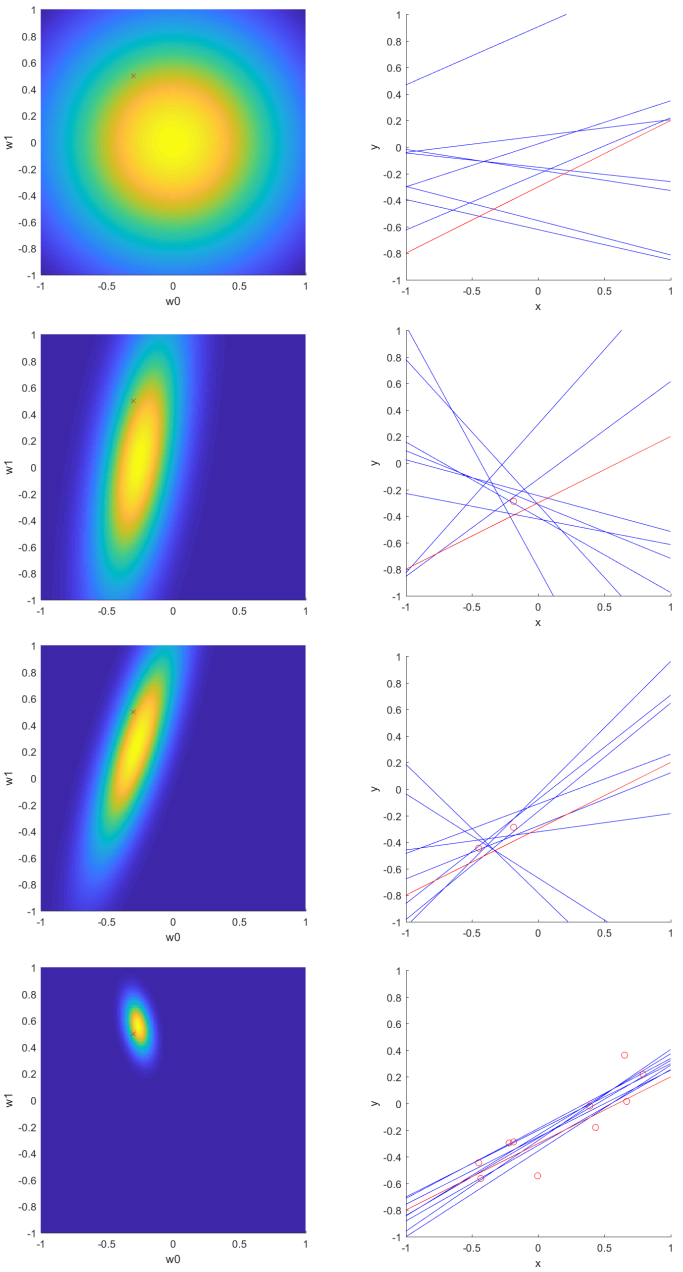

Figure 9: An example of linear regression. Left panel shows the prior/posterior distribution of w. Right panel shows random draws from the prior/posterior. The generating model is in the form of $y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x$, which is the red line in the right panel. An additional Gaussian noise ϵ is added to generate the observations (red circle in the right panel). The red cross in the left panel shows the true value of w_0 and w_1.

the linear coefficients $\boldsymbol{w}$, where $\alpha = 2$ here. We set the noise precision β to the true value $1/0.2^2 = 25$ for simplicity.

As can be seen from the left panel of Figure 9, the posterior distribution becomes more and more confined in a narrow space as more data is observed. Also the whole of the posterior shifts towards the true parameter value (red cross). The right panel show the same idea, in which the red circles are observations. When we only have a vague prior, the lines are randomly scattered in the space. When one data point is observed, all lines pass near this observation. When two data points are observed, the lines are mostly in the correct direction. When 10 points are observed, the sampled lines are almost the same as the generating model (red line).

After parameter inference, the next task we are most interested in is the predictive performance of our linear model. Therefore we derive the following predictive distribution for a new input $\boldsymbol{x}$.

$$p(t|\boldsymbol{t}, \boldsymbol{x}, X, \alpha, \beta) = \int p(t|\boldsymbol{w}, \boldsymbol{x}, \beta) p(\boldsymbol{w}|\boldsymbol{t}, X, \alpha, \beta) d\boldsymbol{w}, \tag{33}$$

where the first term in the integral is the the likelihood of a single observation (see Eq. (27)) and the second term is the posterior of linear coefficients (see Eq. (30)). Since both terms are in the form of Gaussian, there exists a conjugacy and it can be shown that the predictive distribution is also Gaussian

$$p(t|\boldsymbol{t}, \boldsymbol{x}, X, \alpha, \beta) = N(\boldsymbol{m}_N^T \boldsymbol{x}, \sigma_N^2(\boldsymbol{x})), \tag{34}$$

where the mean and variance are given by

$$\boldsymbol{m}_N^T \boldsymbol{x} = \beta \boldsymbol{x}^T \boldsymbol{S}_N X^T \boldsymbol{t} = \sum_{n=1}^{N} \beta \boldsymbol{x}^T \boldsymbol{S}_N \boldsymbol{x}_n t_n \tag{35}$$

$$\sigma_N^2(\boldsymbol{x}) = \frac{1}{\beta} + \boldsymbol{x}^T \boldsymbol{S}_N \boldsymbol{x} \tag{36}$$

A second common task is model comparison, which involves the calculation of the marginal likelihood function or evidence function $p(\boldsymbol{t}|X, \alpha, \beta)$. It can be obtained by integrating out the linear coefficients $\boldsymbol{w}$.

$$p(\boldsymbol{t}|X, \alpha, \beta) = \int p(\boldsymbol{t}|\boldsymbol{w}, X, \beta) p(\boldsymbol{w}|\alpha) d\boldsymbol{w} \tag{37}$$

Due to conjugacy, we can similarly derive the analytic solution of the log marginal likelihood function.

$$\ln p(\boldsymbol{t}|X, \alpha, \beta) = \frac{M}{2} \ln \alpha + \frac{N}{2} \ln \beta - E(\boldsymbol{m}_N) - \frac{1}{2} \ln |A| - \frac{M}{2} \ln(2\pi), \tag{38}$$

where

$$A = \alpha \boldsymbol{I} + \beta X^T X \tag{39}$$

$$\boldsymbol{m}_N = \beta A^{-1} X^T \boldsymbol{t} \tag{40}$$

$$E(\boldsymbol{m}_N) = \frac{\beta}{2} ||\boldsymbol{t} - X\boldsymbol{m}_N||^2 + \frac{\alpha}{2} \boldsymbol{m}_N^T \boldsymbol{m}_N \tag{41}$$

Here we treat the hyper-parameters α and β as fixed. In practice, we would like to tune the hyper-parameters to provide a good fit of the data. There are two general ways to treat the hyper-parameters: 1) maximize the marginal likelihood with respect to the hyper-parameters and 2) impose priors on the hyper-parameters and then integrate out the hyper-parameters. Imposing priors on the hyper-parameters usually leads to a non-conjugate scenario such that integration is difficult. Therefore the first approach is popular due to its simplicity, which is also known as type 2 maximum likelihood. The second approach usually requires MCMC sampling, where the posterior distribution of the hyper-parameters are our target distribution in MCMC. Note that we can calculate the prior and also likelihood analytically given the hyper-parameters. The only thing we do not know is the normalizing constant. The specific procedures of the type 2 maximum likelihood and MCMC sampling are rather complex and we do not provide further introduction here.

2.4.2 Linear regression with basis function

In the previous section, we provide an introduction to Bayesian linear regression, which allows us to model problems with linear effects. In real world applications, we often encounter problems with non-linear effects. Linear models in the previous section will not be able to model these non-linear effects. To tackle this problem, we can use linear model with non-linear basis functions, e.g. polynomials, to model the non-linear effects.

Following the same notation as the previous section, we denote a basis function by $\phi(\boldsymbol{x}) : R^D \to R$, where $\boldsymbol{x} \in R^D$ and $\phi(\boldsymbol{x}) \in R$. Let us assume we have M basis functions in our model. Then each input variable vector $\boldsymbol{x}$ is transformed to a M-dimensional vector $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \cdots, \phi_M(\boldsymbol{x}))^T$. The whole input data matrix X is then transformed to a $N \times M$ design matrix Φ with each element $\Phi_{nj} = \phi_j(\boldsymbol{x}_n)$. The design matrix is shown as follows:

$$\Phi = \begin{pmatrix} \phi_1(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_1) & \cdots & \phi_M(\boldsymbol{x}_1) \\ \phi_1(\boldsymbol{x}_2) & \phi_2(\boldsymbol{x}_2) & \cdots & \phi_M(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{x}_N) & \phi_2(\boldsymbol{x}_N) & \cdots & \phi_M(\boldsymbol{x}_N) \end{pmatrix} \tag{42}$$

Similar to the previous section, we are interested in modelling the target variable as a linear combination of the basis functions and a random Gaussian noise, i.e.

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon, \tag{43}$$

where

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=1}^{M} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}). \tag{44}$$

We can see that the elements of the design matrix Φ plays a similar role as the input data X in a similar manner. If we think the linear regression example of Figure 9 in basis functions, we actually transformed the input variable vector $\boldsymbol{x} = (1, x)^T$ to a vector composed of basis functions $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))^T$, where $\phi_1(\boldsymbol{x}) = 1$ and $\phi_2(\boldsymbol{x}) = [0\,1] \times \boldsymbol{x}$.

We get similar conclusions as the previous section. We place the same prior $p(\boldsymbol{w}|\alpha)$ (Eq. (26)) on the linear coefficients. The likelihood for all target variables is given by

$$p(\boldsymbol{t}|X, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} N(t_n|\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}) = N(\boldsymbol{t}|\Phi \boldsymbol{w}, \beta^{-1} \boldsymbol{I}). \tag{45}$$

The posterior of the linear coefficients $\boldsymbol{w}$ are given by

$$p(\boldsymbol{w}|\boldsymbol{t}, X, \alpha, \beta) = N(\boldsymbol{w}|\boldsymbol{m}_N, \boldsymbol{S}_N), \tag{46}$$

where

$$\boldsymbol{m}_N = \beta \boldsymbol{S}_N \Phi^T \boldsymbol{t} \tag{47}$$

$$\boldsymbol{S}_N^{-1} = \alpha \boldsymbol{I} + \beta \Phi^T \Phi \tag{48}$$

The predictive distribution is

$$p(t|\boldsymbol{t}, \boldsymbol{x}, X, \alpha, \beta) = N(\boldsymbol{m}_N^T \boldsymbol{\phi}(\boldsymbol{x}), \sigma_N^2(\boldsymbol{x})), \tag{49}$$

where the mean and variance are given by

$$\boldsymbol{m}_N^T \boldsymbol{\phi}(\boldsymbol{x}) = \beta \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \Phi^T \boldsymbol{t} = \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}_n) t_n \tag{50}$$

$$\sigma_N^2(\boldsymbol{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}) \tag{51}$$

The log marginal likelihood function is given by

$$\ln p(\boldsymbol{t}|X, \alpha, \beta) = \frac{M}{2} \ln \alpha + \frac{N}{2} \ln \beta - E(\boldsymbol{m}_N) - \frac{1}{2} \ln |A| - \frac{M}{2} \ln(2\pi), \tag{52}$$

where

$$A = \alpha \boldsymbol{I} + \beta \Phi^T \Phi \tag{53}$$

$$\boldsymbol{m}_N = \beta A^{-1} \Phi^T \boldsymbol{t} \tag{54}$$

$$E(\boldsymbol{m}_N) = \frac{\beta}{2} ||\boldsymbol{t} - \Phi \boldsymbol{m}_N||^2 + \frac{\alpha}{2} \boldsymbol{m}_N^T \boldsymbol{m}_N \tag{55}$$

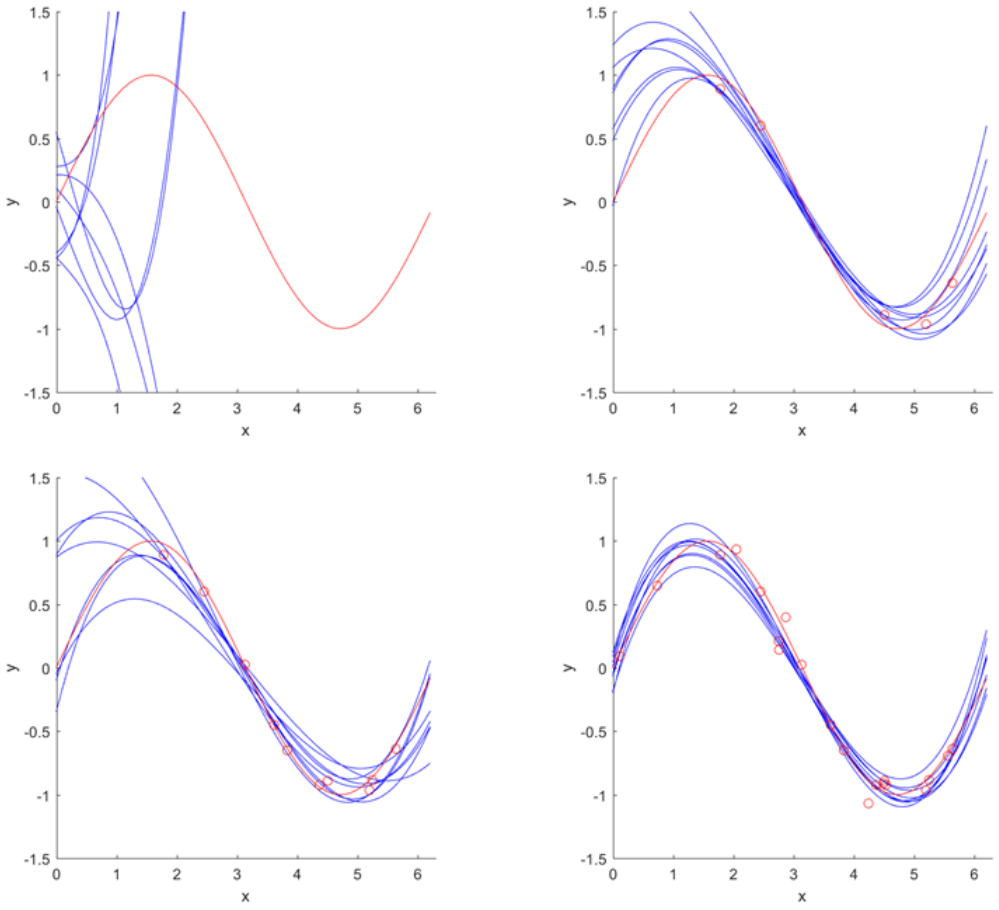

Figure 10: An example of linear regression with polynomial basis function. Top left panel shows random draws from the prior. Top right panel shows random draws from the posterior with 5 observations. The bottom panels show random draws from the posterior with 10 and 20 observations, respectively. The generating model is in the form of $y = sin(x)$, which is the red line in the right panel. An additional Gaussian noise ϵ is added to generate the observations (red circle). The posterior mean of the linear coefficient is $\boldsymbol{m}_N = (-0.0217, 1.6450, -0.7969, 0.0860)^T$.

From the above equations, we can see that linear regression using the basis functions almost identical to the regular linear regression except that X is replaced by Φ. Thus we can use the same set of inference procedures except that we just need to transform the input variables using the basis functions in the beginning. This means we can focus on the choosing proper basis functions to approximate the non-linear effects without worrying about the inference.

Here we show a linear regression example of fitting a sinusoidal curve using polynomial basis functions, as shown in Figure 10. The basis functions chosen for an input variable x is given by $\boldsymbol{\phi}(x) = (1, x, x^2, x^3)^T$. The generating function is given by $t = sin(x) + \epsilon$, where $\epsilon \sim N(0, 0.1^2)$. The prior for linear coefficients $\boldsymbol{w}$ are given by $N(\mathbf{0}, \alpha^{-1}\boldsymbol{I})$, where $\alpha = 2$.

The prior/posterior distributions of $\boldsymbol{w}$ are not shown here since it is difficult to visualize 4-dimensional vectors. We show 8 samples (blue lines) drawn from the priors and posteriors with 5,10,20 observations, respectively. The red lines are the true sinusoidal function and the red circles are the observations. It can be seen that we can better approximate the sinusoidal function as more data are observed. The advantage of non-linear basis function is obvious. Without the polynomial basis functions, we can only fit straight lines. Now we are about to model the non-linearity of sinusoidal function with decent accuracy.

The next question is how to choose the basis functions. There are no general solutions, we need to rely on our experience most of the time. However, we can compare the models easily once the basis functions are identified, using the marginal likelihood function (Eq. (52)). Following the same example in Figure 10, we could fit the data using different subset of basis functions: $\boldsymbol{\phi}_1(x) = (1, x)^T$, $\boldsymbol{\phi}_2(x) = (1, x, x^2)^T$, $\boldsymbol{\phi}_3(x) = (1, x, x^2, x^3)^T$ and $\boldsymbol{\phi}_3(x) = (1, x, x^2, x^3, x^4)^T$, corresponding to models denoted by M_1, M_2, M_3 and M_4, respectively. Figure 11 shows the fitting of the different models and their corresponding log marginal likelihood, where the blue lines are random draws from the posterior. The log marginal likelihood (Eq. (52)) of the 4 models are -62.9, -67.7, 49.4, 43, respectively. It can be seen that model M_3 and M_4 fit the data best. M_1 and M_2 are not flexible enough to model the non-linearity in the data. M_3 fits well and has higher log marginal likelihood than M_4. This means M_3 is simple while still provide a nice fit to the data, thus it is the best model. The log Bayes factor of M_3 over M_4 is 49.4-43=6.4, which shows a higher preference for M_3.

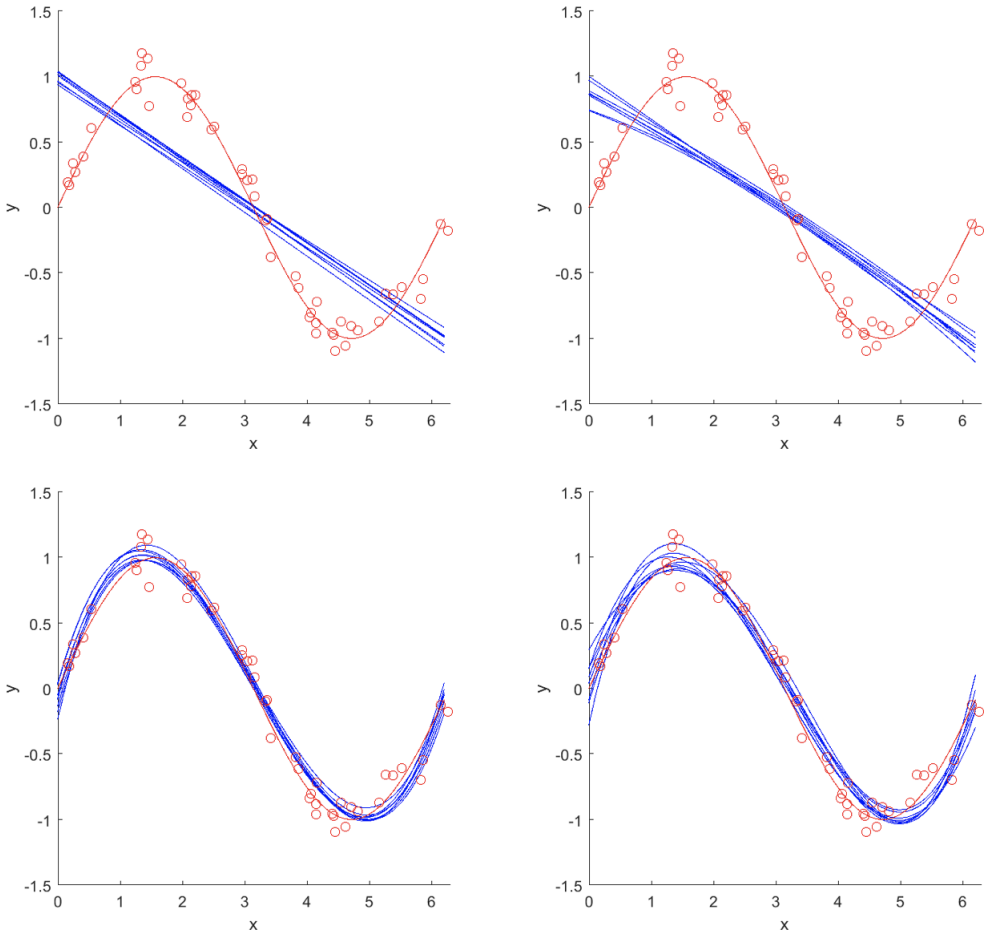

Figure 11: Model comparison. From top to bottom, from left to right, the four models (M_1, M_2, M_3, M_4) fit the same data using linear regression with basis functions $(1, x)$,$(1, x, x^2)$,$(1, x, x^2, x^3)$,$(1, x, x^2, x^3, x^4)$, respectively. The real data are generated using $y = sin(x)$ (red line). An additional Gaussian noise ϵ is added to generate the 50 observations (red circle). The blue lines are samples from the posterior distribution. The log marginal likelihood (Eq. (52)) is -62.9, -67.7, 49.4, 43, respectively.

2.4.3 Gaussian process regression

In the previous two sections, we have seen that we can sample functions by sampling linear coefficient $\boldsymbol{w}$ from its distribution. The sampled functions are linear combinations of the basis functions, with their weights given by $\boldsymbol{w}$. So in a way we are performing Bayesian inference in function space. We can think we have a prior full of different random functions. When data are observed, the posterior converged to a distribution with random functions of constrained shapes by the observations.

The idea of performing Bayesian inference in the function space turns out to be a Gaussian process, which provide a nice way to govern the functions.

First let us look at the predictive mean (Eq. (50)) in the linear regression with basis functions.

$$y(\boldsymbol{x}, \boldsymbol{m}_N) = \boldsymbol{m}_N^T \boldsymbol{\phi}(\boldsymbol{x}) = \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}_n) t_n, \tag{56}$$

where $\boldsymbol{x}$ is a new observation, $\boldsymbol{m}_N$ and $\boldsymbol{S}_N$ are posterior mean and variance. It can be seen the predictive mean is a linear combination of the target variables in the training data.

$$y(\boldsymbol{x}, \boldsymbol{m}_N) = \sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_n) t_n, \tag{57}$$

where $k(\boldsymbol{x}, \boldsymbol{x}')$ is defined as the equivalent kernel function

$$k(\boldsymbol{x}, \boldsymbol{x}') = \beta \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}') \tag{58}$$

It can been seen that the equivalent kernel function purely depends on the inner product of $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\phi}(\boldsymbol{x}_n)$, where $n \in (1, 2, \cdots, N)$. A next idea is whether we can conduct our Bayesian linear regression purely by the inner product of our basis functions. It turns out to be possible if we define the following kernel function.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x}') \tag{59}$$

We are interested in the form of the linear combination of the basis functions, which is actually $y(\boldsymbol{x}, \boldsymbol{w})$. We denote the function values over all input variables by $\boldsymbol{y}$, where $y_n = y(\boldsymbol{x}_n, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}_n$. It can be written in the following form

$$\boldsymbol{y} = \Phi \boldsymbol{w}, \tag{60}$$

where Φ is the design matrix defined in Eq. (42) and $\Phi_{nk} = \phi_k(\boldsymbol{x}_n)$. Since $\boldsymbol{w}$ is a multivariate Gaussian distribution, the linear combination of it is also a

Gaussian, which means $\boldsymbol{y}$ is in a Gaussian form. The mean and variance are given by

$$E[\boldsymbol{y}] = \Phi E[\boldsymbol{w}] = \mathbf{0} \tag{61}$$

$$Cov[\boldsymbol{y}] = E[\boldsymbol{y}\boldsymbol{y}^T] = \Phi E[\boldsymbol{w}\boldsymbol{w}^T]\Phi^T = \frac{1}{\alpha}\Phi\Phi^T = K, \tag{62}$$

where K is the Gram matrix with elements

$$K_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \frac{1}{\alpha}\boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{\phi}(\boldsymbol{x}_m) \tag{63}$$

We can write the prior of the random function values over input data in the following form

$$p(\boldsymbol{y}|X) = N(\mathbf{0}, K) \tag{64}$$

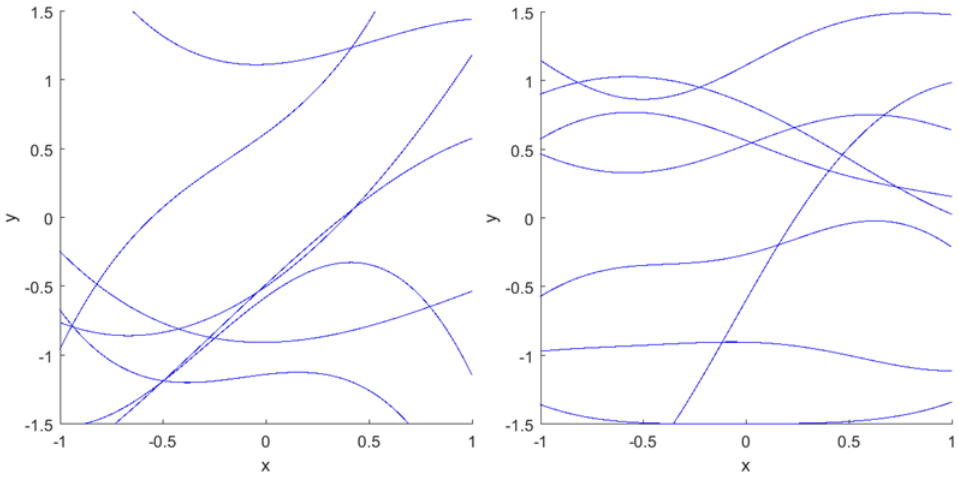

Figure 12: Randomly sampled functions from Gaussian process. The left panel shows 8 random functions sampled from the GP using the kernel (Eq. (66)) determined by the polynomial basis functions in the previous section. The right panel shows 8 random functions drawn from GP using the squared exponential kernel (Eq. (65)) with $\sigma^2 = 1$ and $l = 1$.

Now we can sample values of $\boldsymbol{y}$ directly from multivariate normal distribution $N(\mathbf{0}, K)$ without sampling $\boldsymbol{w}$, as long as the Gram matrix is given. This means we can shift from specifying the basis function to modeling the covariance function (or Gram matrix), where we only need to specify the kernel function. The kernel function implicitly encodes the inner product of basis functions. Specifying the kernel function brings us great flexibility since we can use a kernel that corresponds to infinite basis functions. A commonly

used kernel that corresponds to infinite basis functions [14] is the squared exponential kernel given by

$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right), \tag{65}$$

where σ^2 is the magnitude parameter and l is the length scale parameter. σ^2 and l are regarded as fixed hyper-parameters. This kernel imposes an infinitely differentiable property on its sampled functions, i.e. the sampled functions are smooth everywhere. The kernel function corresponds to the polynomial basis function $\boldsymbol{\phi}(x) = (1, x, x^2, x^3)^T$ in the previous section is given by

$$k(x, x') = \frac{1}{\alpha}\boldsymbol{\phi}(x)^T\boldsymbol{\phi}(x') = \frac{1}{\alpha}(1 + xx' + x^2x'^2 + x^3x'^3) \tag{66}$$

Figure 12 shows the sampled functions from both kernels. The left and right panel show random functions drawn from Gaussian process using the kernels defined in Eq. (66) and Eq. (65), respectively.

Once we have specified the kernel, we implicitly impose a prior on the random functions. So Gaussian process can be viewed as a distribution over functions. Next we introduce how to perform Bayesian inference under the Gaussian process framework.

Similar to the previous section, we add i.i.d. Gaussian noises to the random function to generate the observations. Figure 13 shows the graphic model of Gaussian process. The joint distribution of the target variables, or the likelihood, is given by

$$p(\boldsymbol{t}|\boldsymbol{y}) = N(\boldsymbol{t}|\boldsymbol{y}, \beta^{-1}\boldsymbol{I}_N), \tag{67}$$

where β is the precision of Gaussian noise and $\boldsymbol{I}_N$ is the $N \times N$ identity matrix.

When deriving the posterior distribution of the random functions $\boldsymbol{y}$, there is some difference in the mechanism compared with normal Bayesian inference of the parameters, such as $\boldsymbol{w}$ in the linear regression case. The difference lies in the fact that the prior of $\boldsymbol{y}$ depends on the input data X through the Gram matrix K, while the prior of $\boldsymbol{w}$ is independent of X. This specialty requires us to specify the input data X^* beforehand whenever we want to check the posterior distribution of the random function over these input values. In some sense, the posterior distribution over the random functions is actually the predictive distribution of new input data X^*.

Since the prior (Eq. (64)) and the likelihood (Eq. (67)) are both Gaussian, the marginal is also Gaussian given by

$$p(\boldsymbol{t}) = \int p(\boldsymbol{t}|\boldsymbol{y})p(\boldsymbol{y}) = N(\boldsymbol{t}|\boldsymbol{0}, C_N), \tag{68}$$

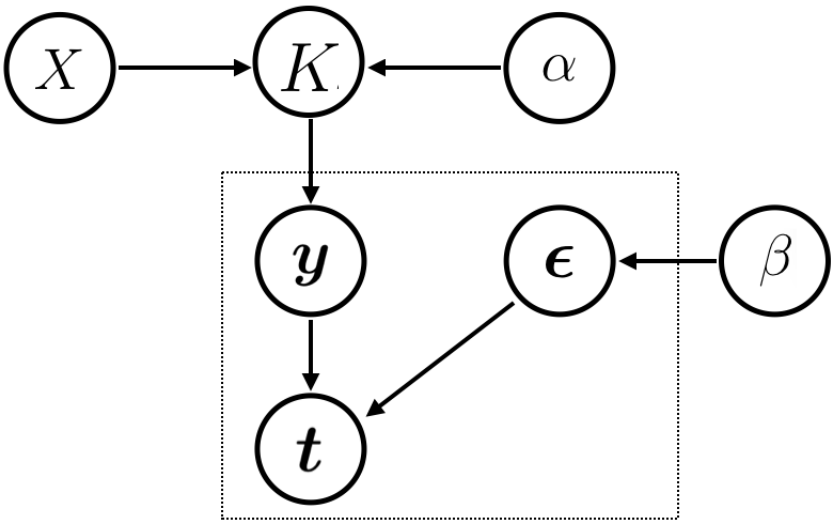

Figure 13: Graphical model of Gaussian process. Since we model the whole set of training data all together, there are not repetitions in the dashed rectangle. α refers to general hyper parameters for the adopted kernel and β refers to general hyper parameters for the noise.

where C_N is a $N \times N$ covariance matrix and its element is given by

$$C(\boldsymbol{x}_n, \boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \beta^{-1}\delta_{nm} \tag{69}$$

and $\delta_{nm} = 1$ only when $n = m$, otherwise it is 0.

To derive the predictive distribution $p(t_{N+1}|\boldsymbol{t})$ for a new input data $\boldsymbol{x}_{N+1}$, we note the joint distribution of $\boldsymbol{t}_{N+1} = (t_1, t_2, \cdots, t_N, t_{N+1})^T$ follows a similar form as Eq. (68) given by

$$p(\boldsymbol{t}_{N+1}) = N(\boldsymbol{t}_{N+1}|\boldsymbol{0}, C_{N+1}), \tag{70}$$

where the covariance matrix can be written as

$$C_{N+1} = \begin{pmatrix} C_N & \boldsymbol{k} \\ \boldsymbol{k}^T & k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) + \beta^{-1} \end{pmatrix}, \tag{71}$$

and the vector $\boldsymbol{k}$ has the elements $k(\boldsymbol{x}_n, \boldsymbol{x}_{N+1})$ for $n = 1, \ldots, N$. Using the conclusion of Gaussian conditional distribution [3], we can derive the predictive distribution $p(t_{N+1}|\boldsymbol{t})$

$$p(t_{N+1}|\boldsymbol{t}) = N(t_{N+1}|m(\boldsymbol{x}_{N+1}), \sigma^2(\boldsymbol{x}_{N+1})), \tag{72}$$

where

$$m(\boldsymbol{x}_{N+1}) = \boldsymbol{k}^T C_N^{-1} \boldsymbol{t} \tag{73}$$

$$\sigma^2(\boldsymbol{x}_{N+1}) = k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) + \beta^{-1} - \boldsymbol{k}^T C_N^{-1} \boldsymbol{k} \tag{74}$$

The predictive distribution of the target variables $\boldsymbol{t}^*$ for multiple input data points X^* has a similar form as follows.

$$p(\boldsymbol{t}^*|\boldsymbol{t}) = N(\boldsymbol{t}^*|m(X^*), \sigma^2(X^*)), \tag{75}$$

where

$$m(X^*) = \boldsymbol{k}^T C_N^{-1} \boldsymbol{t} \tag{76}$$

$$\sigma^2(X^*) = k(X^*, X^*) + \beta^{-1} \boldsymbol{I} - \boldsymbol{k}^T C_N^{-1} \boldsymbol{k} \tag{77}$$

$$\boldsymbol{k} = k(X, X^*) \tag{78}$$

The predictive distribution of the latent functions $\boldsymbol{y}^*$ for multiple input data points X^* are given by.

$$p(\boldsymbol{y}^*|\boldsymbol{t}) = N(\boldsymbol{y}^*|m(X^*), \sigma^2(X^*)), \tag{79}$$

where

$$m(X^*) = \boldsymbol{k}^T C_N^{-1} \boldsymbol{t} \tag{80}$$

$$\sigma^2(X^*) = k(X^*, X^*) - \boldsymbol{k}^T C_N^{-1} \boldsymbol{k} \tag{81}$$

$$\boldsymbol{k} = k(X, X^*) \tag{82}$$

We present an example of the Gaussian process predictive distribution. First we simulate 7 data points using the same approach as Figure 10, next we derive the predictive distribution on test data which densely cover the input space, and finally we draw 8 samples from the predictive distribution. Figure 14 shows the predictive distributions using the polynomial kernel and squared exponential kernel for the same data. We can see both kernels fit the data very well.

We notice that the regression inference is almost "deterministic" if the hyper-parameters (α in Eq. (66) and l, σ in Eq.(65)) are given. There are no parameters that needs to be estimated. The next step is to treat the hyper parameters as unknown, and try to estimate the hyper parameters. We can either maximize the marginal likelihood (Eq. (68)) to get point estimates of the hyper parameters, or place an prior on them and infer their posterior MCMC. Due to the technical complexity of the optimization and MCMC, we do not provide further details.

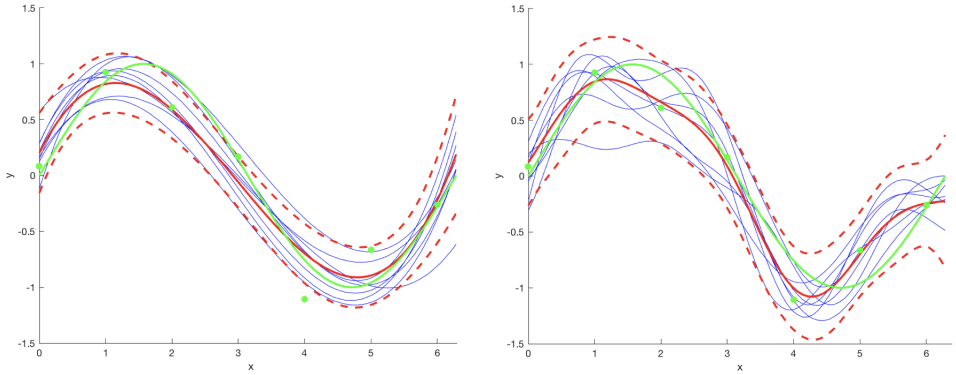

Figure 14: Randomly sampled functions from Gaussian process posterior. The left panel shows 8 random functions sampled from the GP posterior using the kernel (Eq. (66)) determined by the polynomial basis functions in the previous section. The right panel shows 8 random functions drawn from GP posterior using the squared exponential kernel (Eq. (65)) with $\sigma^2 = 1$ and $l = 1$. The green lines and dots are the generating function $y = sin(x)$ and the observations. The central solid red line is the posterior mean. The dashed red lines are the posterior mean ± 2 standard deviations. The blue lines are random samples drawn from the posterior.

3 Methods and Results

We will introduce the specific data and methods of our modelling in this section. Then we present the results by applying different methods on the data.

3.1 Data

The data used for this thesis is internal T1D dataset from my colleague Juhi Somani. It contains 49386 gene expression data (microarray), or intensities, for 10 case-control pairs. Each case-control pair is matched by age, gender, genetic background. For each individual (case or control), blood samples are taken at similar ages. Cases refer to individuals with sero conversion at some time point of their time. Figure 15 shows the sample collection time of each case aligned to the sero conversion event. Since we are interested in discovering the effects brought by sero conversion, we need data both before and after the sero conversion. From the 10 case-control pairs, we further selected 6 pairs 2,3,7,8,9,10, which meet this criterion. The gene expression data (intensities) are log2-transformed in the preprocessing steps.

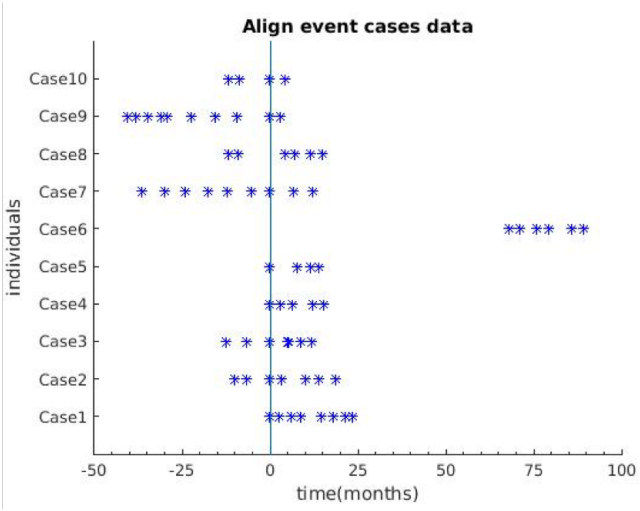

Figure 15: Sample collection time points of cases aligned to sero conversion. y-axis is the case id. x-axis is the time to sero conversion. Blue stars are the sample collection time points. As can be seen, pairs 2,3,7,8,9,10 have data points both before and after the sero conversion.

3.2 Methods

We are interested in discovering genes that differs between the cases and controls. The general idea is that if there is no difference, then both cases and controls can be described by a shared model, otherwise they should deviate from the shared model.

Here we focus on modeling only one gene for one matched case-control pair. Let us denote the gene expression data or observations by $\boldsymbol{t} = (t_1, \ldots, t_N)^T$ and the age at sample collection by $\boldsymbol{x} = (x_1, \ldots, x_N)^T$.

3.2.1 Standard linear regression

A straightforward idea is to assume that 1) all data points of the case-control pair are generated by the same linear model and 2) the sero conversion event causes some abnormality that leads to outliers which are unlikely to be generated by the shared linear model. The linear model is given by

$$t = w_1 + w_2 x + \epsilon, \tag{83}$$

where $\boldsymbol{w} = (w_1, w_2)^T$ is the linear coefficients and $\epsilon \sim N(0, \sigma^2)$ is $i.i.d$ noise.

We can easily obtain the maximum likelihood estimates of $\boldsymbol{w}$ and σ^2 using standard frequentist linear regression software packages. The procedures are very similar to the Bayesian linear regression in section 2.4.1 except that here we get point estimates $\hat{\boldsymbol{w}}$ and $\hat{\sigma}^2$.

Outliers can significantly affect the obtained maximum likelihood estimates and make the outlier detection difficult if we simply fit the model to the whole dataset. We use the following strategy to alleviate this problem. We take one data point (x_i, t_i) out at a time, then fit the linear model to the rest data points $(\boldsymbol{x}_{-i}, \boldsymbol{t}_{-i})$. Once the maximum likelihood estimates $\hat{\boldsymbol{w}}$ and $\hat{\sigma}^2$ are ready, we can obtain the predictive distribution of x_i, which is given by

$$y_i = N(y_i | \hat{w}_1 + \hat{w}_2 x_i, \hat{\sigma}^2). \tag{84}$$

We then calculate the p-value of the actual observation t_i given this predictive distribution. After we obtain the p-values for all the data points, we perform FDR corrections and report outliers that reach the predefined significance level.

3.2.2 Bayesian linear regression

One problem of the frequentist linear regression is that the method is relatively "rigid" and can easily provide false positive predictions. We want to make the

predictive distribution of y_i a little bit more flexible, such that the line can rotate a little bit and tolerate more "outliers".

To achieve this, we want to treat the mean and variance of $\boldsymbol{y}_{-i}$ as random variables and place a prior on them. This problem can be summarized as a multivariate Gaussian distribution with unknown mean and variance and we can use the Normal-inverse-Wishart distribution (Eq. 10) as the prior. In our case, we just need to replace the observations $\boldsymbol{x}$ in Eq. 11 with $\boldsymbol{t}_{-i}$ and $n = 1$ here.

Once we have obtained the posterior distribution of the mean and variance for $\boldsymbol{y}_{-i}$, we then draw 1000 samples from this distribution by the following two-step procedures.

$$(\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}) \sim NIW(\boldsymbol{\mu}', \kappa', \nu', \Psi') \tag{85}$$

$$\boldsymbol{y}_{-i}^{(k)} \sim N(\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}), \tag{86}$$

where $k = 1, \ldots, 1000$.

For each sampled $\boldsymbol{y}_{-i}^{(k)}$, we can obtain the maximum likelihood estimates $\hat{\boldsymbol{w}}^{(k)}$ and $\hat{\sigma}^{2(k)}$ following the same procedures in the previous section. Given $\hat{\boldsymbol{w}}^{(k)}$ and x_i, next we calculate the predictive value $\hat{y}_i^{(k)} = \hat{w}_1^{(k)} + \hat{w}_2^{(k)} x_i$.

At this stage we have 1000 estimates of $\hat{y}_i^{(k)}$, we then fit a Gaussian distribution to the 1000 estimates and calculate the p-value for t_i using the fitted Gaussian distribution. In this way we calculate the p-values repetitively for all data points, after which we perform FDR correction to the p-values and report the significant hits.

3.2.3 Gaussian process regression

Since the data exhibit a nonlinear property, we can model the nonlinearity using Gaussian process. The idea follows closely to the previous two subsections. We want to obtain the predictive distribution of y_i, based on which we then calculate the p-value of t_i.

Here we use squared exponential kernel to model the nonlinearity. As shown in section 2.4.3, the predictive distribution of y_i is given by Eq. 79, which is a Gaussian distribution given by

$$p(y_i|\boldsymbol{t}_{-i}) = N(y_i|m(\boldsymbol{x}), \sigma^2(\boldsymbol{x})), \tag{87}$$

where

$$m(\boldsymbol{x}) = \boldsymbol{k}^T C_{N-1}^{-1} \boldsymbol{t}_{-i} \tag{88}$$

$$\sigma^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^T C_{N-1}^{-1} \boldsymbol{k}. \tag{89}$$

The elements of C_{N-1} and $\boldsymbol{k} = (k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_N)^T$ are given by

$$C_{nm} = k(x_n, x_m) \tag{90}$$
$$k_n = k(x_n, x_i) \tag{91}$$

where $m, n \in (1, \ldots, i-1, i+1, \ldots, N)$.

The hyperparameters l, σ^2 and σ_ϵ^2 of the squared exponential kernel and the noise remain to be estimated. We initialize them to different values and then optimize the marginal likelihood to decide the final point estimates of the hyper parameters.

Given the point estimates of the hyperparameters, we easily derive the predictive distribution of y_i and calculate the p-value of t_i. After that we do FDR correction and report the significant hits.

3.2.4 Gaussian process model comparison

In the previous three subsections, we focus on deriving a predictive distribution for each data point and decide whether it is an outlier by checking its p-value. This section try to model this problem from a different perspective. There may exist systematic change between the case and control, e.g. the target value of the case is always larger than the control. Outlier modeling is not able to detect this kind of difference.

We propose to model the case-control pair using a shared GP model M_S and a independent GP model M_I. The shared GP models case and control jointly using shared parameters, while the independent GP models the case and control separately. If there is not much difference, then the shared model will be preferred, otherwise the independent model is preferred. We use squared exponential kernel for the both models.

We first present the shared model M_S, where we make no difference of the case and the control. So it is simply a Gaussian process regression over all data points of the case-control pair and the marginal likelihood is given by Eq.(68).

We fit a separate GP for the case and the control in the independent model M_I. Let us denote the data points by $\boldsymbol{x} = (\boldsymbol{x}_{case}^T, \boldsymbol{x}_{control}^T)^T$, where $\boldsymbol{x}_{case}$ are data points belonging to the case and $\boldsymbol{x}_{control}$ are those belonging to the control. Similarly we use $\boldsymbol{t} = (\boldsymbol{t}_{case}^T, \boldsymbol{t}_{control}^T)^T$ to denote the target variables. The marginal likelihood of M_I is given as follows, which is the product of the marginal likelihood of the GP fitted to the ($\boldsymbol{x}_{case}$,$\boldsymbol{t}_{case}$) and ($\boldsymbol{x}_{control}$,$\boldsymbol{t}_{control}$), respectively.

$$p(\boldsymbol{t}|\boldsymbol{x}, M_I) = p(\boldsymbol{t}_{case}|\boldsymbol{x}_{case}) p(\boldsymbol{t}_{control}|\boldsymbol{x}_{control}), \tag{92}$$

where $p(\boldsymbol{t}_{case}|\boldsymbol{x}_{case})$ and $p(\boldsymbol{t}_{control}|\boldsymbol{x}_{control})$ are given by Eq.(68). Note that we have omitted the hyperparameters for simplicity. Both the case and the control have an independent set of hyperparameters.

After obtaining the point estimates of the hyperparameters, we can then check the log Bayes factor between the shared model and the independent model.

$$\ln BF = \ln p(\boldsymbol{t}|\boldsymbol{x}, M_S) - \ln p(\boldsymbol{t}|\boldsymbol{x}, M_I) \tag{93}$$

If the log Bayes factor is greater than 0, then the shared model is preferred. If the log Bayes factor is less than 0, then the independent model is preferred and it is a hint of case control difference.

3.3 Results

Since the data is not published yet, we will not show all the results. We first provide a brief summary of the results. Then we show some example results by applying the methods introduced in the previous sections to the data. We focus more on patterns that look interesting rather than making biological interpretations.

For the first three methods, we get p-values for 49386 genes of 6 pairs, each of which contains several time points. We then performed FDR correction of all the p-values using both Bonferroni correction and Benjamini-Hochberg correction. We set the p-value significance threshold to be 0.05. After that we count significant genes, which have at least a single detected outlier in any pair. The total number of significance genes is 4956, 661 and 2797 for the three methods using Bonferroni correction. The numbers are 43276, 3584 and 25149 after Benjamini-Hochberg correction. For the last GP model comparison method, we calculate the log Bayes factor of the shared model versus the independent model (Eq.(93)) for 49386 genes of 6 pairs. We select significant genes by requiring the log Bayes factor to be less than -5 in at least 1 pair, i.e. independent model is preferred. In total we get 722 significant genes.

Now we show some examples of the significant results.

First we show a significant result of the standard linear regression in Figure 16. We use the Matlab function `fitlm()` for the linear regression. The red solid line shows the linear fit to the training data and the dashed red lines show ± 2 standard deviation confidence interval. The data point taken out, or the test data, is marked as red star and the nearby number is the p-value. As can be seen from Figure 16, the outlier (red star) do differs from the rest data points, therefore the method works as expected. The significance level may not be really high since the outlier is still close to other data points.

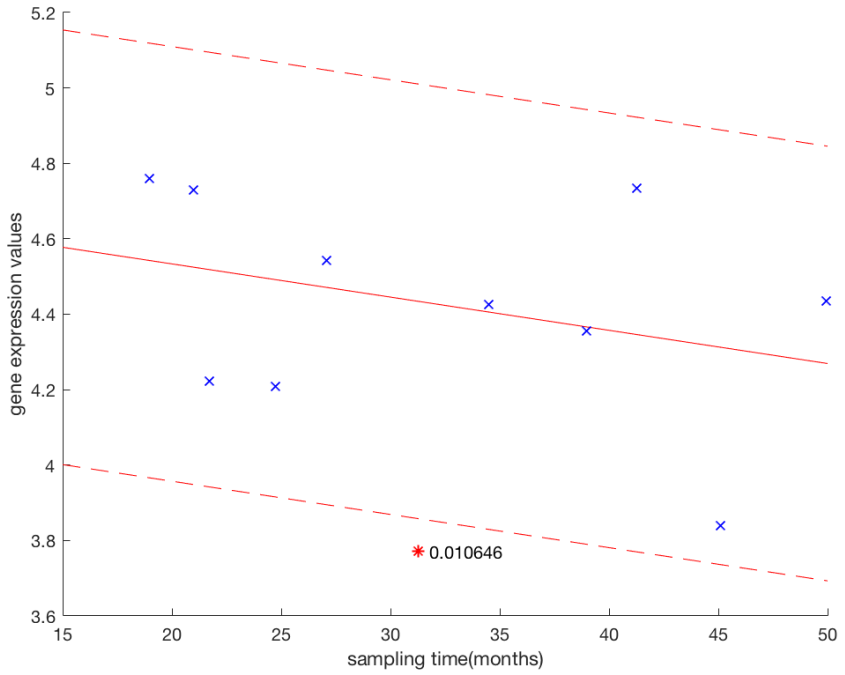

Figure 16: A significant gene with outlier detected by standard linear regression. y-axis is the gene intensity. x-axis is the blood sample collection time. The data is from case-control pair 2. The training data points marked as blue cross. The outlier is marked as red star, which has a p-value 0.0106 given nearby. The solid red show the linear fit and the dashed red lines shows the confidence interval of ± 2 standard deviation.

Next we show the example of the outlier detection using Bayesian linear regression applied to the same data, as shown in Figure 17. We randomly sample 4 mean and variance values from the posterior, each of which corresponds to a different linear fit. Then the mean and confidence intervals are shown for each sampled mean and variance using the same method as in Figure 9. We use the following hyperparameters for the Normal-inverse-Wishart prior distribution: $\boldsymbol{\mu}_0 = 3.6 \times \mathbf{1}$, $\kappa_0 = 2$, $\nu_0 = 10$ and $\Psi = 0.3\boldsymbol{I}$. As can be seen from Figure 17, the variance of the sampled linear fit from the posterior is really high, which leads to a high p-value for the taken out data point compared with that in Figure 17. We have changed the prior to several different values and still observe the same high variance. We think it is because we only have one observation for the prior, which is not enough to converge to a shrinked posterior. Although with the problem of high variance, the method is less sensitive to report outliers, which is expected.

Figure 18 shows an example of outlier detection using Gaussian process regression. The biggest difference is that we are able to fit a nonlinear curve to the data, which provides more flexibility. We initialize the hyperparameters $(l, \sigma^2, \sigma_\epsilon^2)$ to (0.4,0.4,20), (1,3,70), (3,5,120). Then we optimize the hyperparameters with respect to the marginal likelihood separately and choose the largest out of this three.

In the end we show two interesting examples of applying the Gaussian process model comparison method. We use `GPstuff` [18] to implement this method. For the length scale parameter l, we use a t-distribution prior with hyperparameters $\mu = 0$, $\sigma^2 = 100$ and $\nu = 10$. For the magnitude parameter σ^2, we use a square root t-distribution prior with hyperparameters $\mu = 0$, σ^2=300 and $\nu = 4$. We use a log uniform prior for the noise variance σ_ϵ^2.

Figure 19 and 20 show two interesting examples of genes that show difference between the case and control. As can be seen from Figure 19, the case has large variations while the control has relatively low variation. In Figure 20, we can see that the gene intensity of the case is systematically higher than the control.

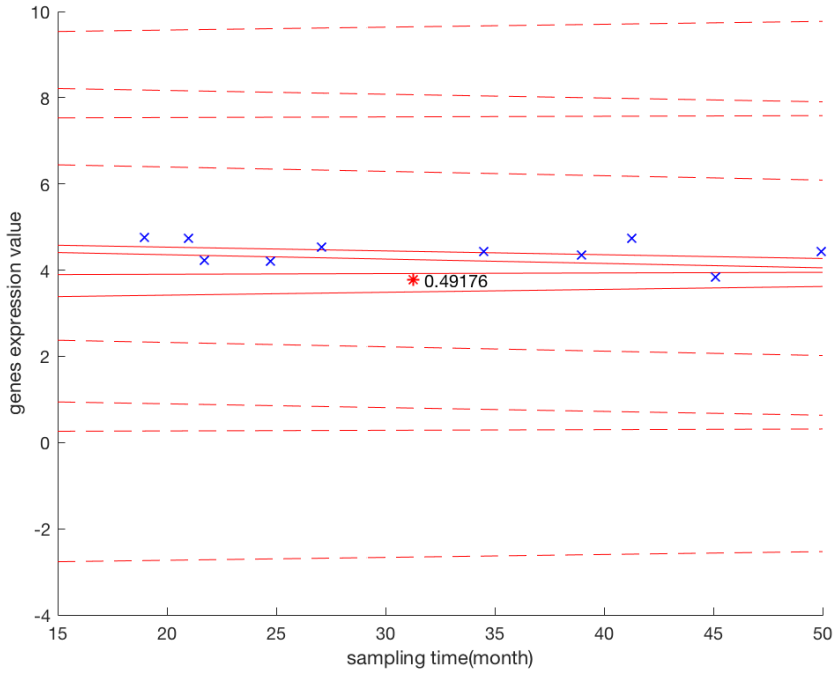

Figure 17: Bayesian linear regression applied to the same data as Figure 16. y-axis is the gene intensity. x-axis is the blood sample collection time. The training data points marked as blue cross. The taken out data point is marked as red star, which has a p-value 0.49176 shown nearby. The solid red lines show the linear fit and the dashed red lines shows the confidence intervals of $\pm\mathbf{x}$ standard deviation, for 4 randomly sampled mean and variance from the posterior.

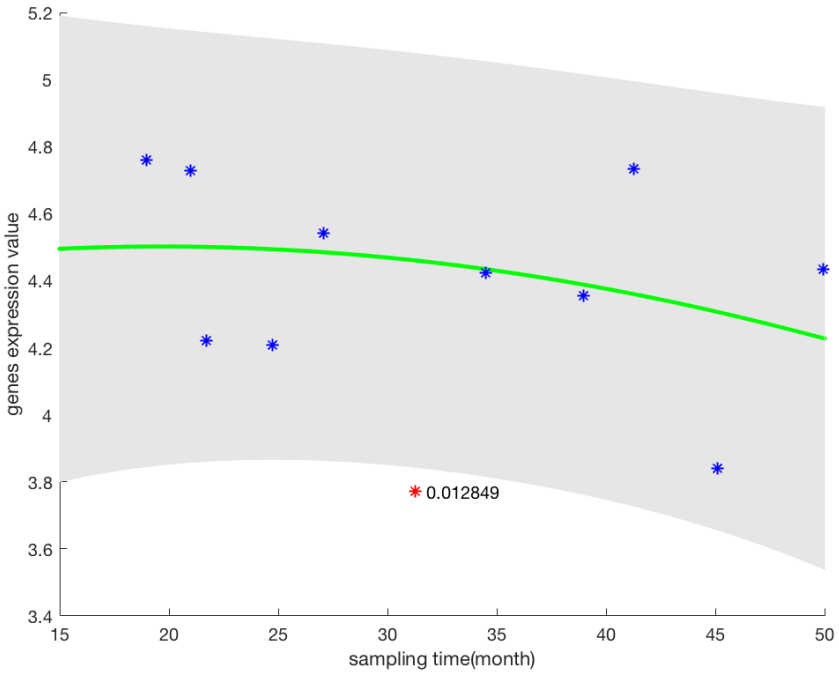

Figure 18: Gaussian process regression applied to the same data as Figure 16. y-axis is the gene intensity. x-axis is the blood sample collection time. The training data points marked as blue stars. The taken out data point, or the test data, is marked as red star, with a p-value 0.0128 shown nearby. The solid green line shows the GP predictive mean and shaded area shows the confidence interval of ± 2 standard deviation.

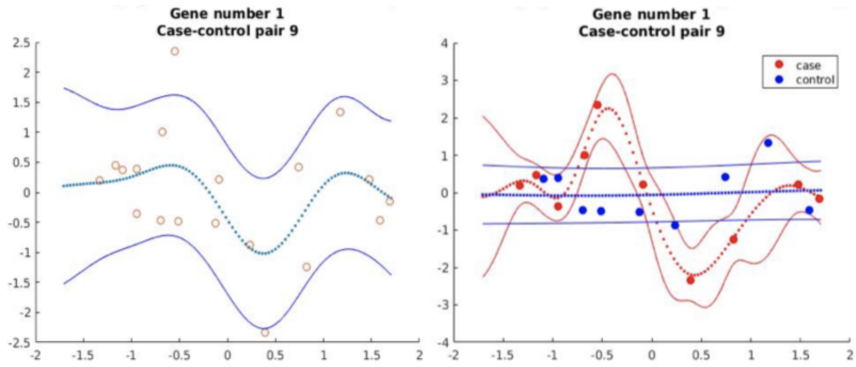

Figure 19: An example of Gaussian process model comparison. y-axis is gene intensity and x-axis is age, both are shown in the normalized scale. The data is coming from a gene of case-control pair 9. The left panel shows the GP fitting using the shared model, where the cyan line is the posterior mean and the blue lines are the ± 2 standard deviation confidence interval. The right panel shows the GP fitting to the case and control separately, where the case is marked as red and the control is marked as blue.

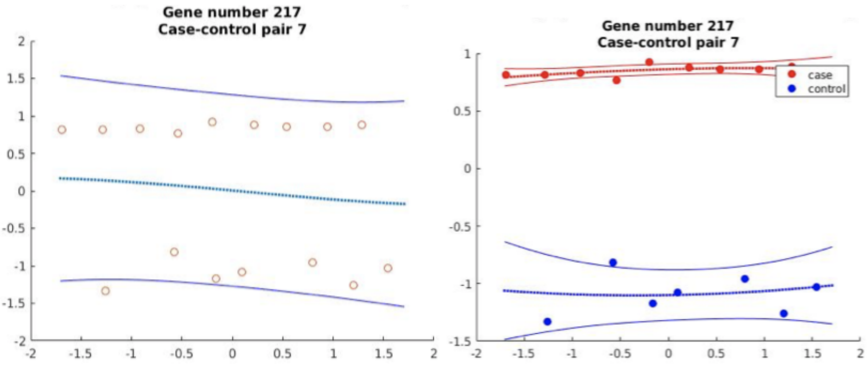

Figure 20: An example of Gaussian process model comparison. y-axis is gene intensity and x-axis is age, both are shown in the normalized scale. The data is coming from a gene of case-control pair 7. The left panel shows the GP fitting using the shared model, where the cyan line is the posterior mean and the blue lines are the ± 2 standard deviation confidence interval. The right panel shows the GP fitting to the case and control separately, where the case is marked as red and the control is marked as blue.

4 Discussion

In this thesis, we have presented 4 statistical methods for analyzing T1D gene expression data. They successfully identify the patterns we want to discover from the data, which exhibit the difference between cases and controls. These differences can be easily seen from the example significant results in the previous section. These methods are able to detect outliers and systematic difference between cases and controls. We can easily observe that Gaussian process can fit the data much better than linear regression.

Discovering interesting biomarkers for T1D prediction is more difficult. We can see that there are lots of significant hits from the result summary. We can do some pathway enrichment analysis to understand the results better in further analysis. The Bonferroni correction is much more strict than the Benjamini-Hochberg correction, which leads to less number of significant hits. Given the large number of significant hits, it seems to be challenging to select the potential biomarkers. We need to validate the top hits using extra biological experiments.

The numbers of significant hits between the first three methods and the last method are not comparable. One reason is that the last method is checking the systematic difference, while the first three methods are checking single outlier. Another reason there is no obvious one to one correspondence between the p-value and log Bayes factor significance level.

The Bayesian linear regression method is more flexible than standard linear regression, which leads to less significant counts. The results hints that the standard linear regression is more sensitive to outliers and may likely to report more false positives. It may be a good idea to use posterior distribution rather than point estimates for the parameters.

Following the same idea, it will be beneficial if we can place a prior on the parameters (length scale, magnitude and noise) in the Gaussian process regression model and derive its posterior in the fourth method. Then we can integrate out the parameters to gain a more robust conclusion. There are no known conjugate priors for these parameters, so we have to use MCMC. With the posterior samples, we can get an expected marginal likelihood. This will be a nice idea for future development.

Another idea is that the cases may only exhibit difference during certain period. This brings rapid changes to some genes. Our GP model may not be able to capture this kind of phenomenon since it only model smooth functions with the same length scale. It may worth thinking about non-stationary Gaussian process models for this kind of phenomenon. Then we can test if cases differs from controls in a small time window.

References

[1] Atkinson, Mark A.: *The pathogenesis and natural history of type 1 diabetes*. Cold Spring Harbor Perspectives in Medicine, 2(11), 2012.

[2] Benjamini, Yoav and Hochberg, Yosef: *Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing*. Journal of the Royal Statistical Society. Series B (Methodological), 57(1):289–300, 1995, ISSN 00359246. http://dx.doi.org/10.2307/2346101.

[3] Bishop, Christopher M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006, ISBN 0387310738.

[4] Brooks, Steve, Gelman, Andrew, Jones, Galin, and Meng, Xiao Li: *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

[5] Elkon, Keith and Casali, Paolo: *Nature and functions of autoantibodies*. Nature Clinical Practice Rheumatology, 4:491–498, September 2008.

[6] Filippi, Christophe M. and Herrath, Matthias G. von: *Viral trigger for type 1 diabetes*. Diabetes, 57(11):2863–2871, 2008, ISSN 0012-1797. http://diabetes.diabetesjournals.org/content/57/11/2863.

[7] Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., and Rubin, D.B.: *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013, ISBN 9781439840955. https://books.google.co.uk/books?id=ZXL6AQAAQBAJ.

[8] Greenland, Sander, Senn, Stephen J., Rothman, Kenneth J., Carlin, John B., Poole, Charles, Goodman, Steven N., and Altman, Douglas G.: *Statistical tests, p values, confidence intervals, and power: a guide to misinterpretations*. European Journal of Epidemiology, 31(4):337–350, Apr 2016, ISSN 1573-7284. https://doi.org/10.1007/s10654-016-0149-3.

[9] Grimmett, Geoffrey and Stirzaker, David: *Probability and random processes*. Oxford University Press, Oxford; New York, 2001, ISBN 0198572239 9780198572237 0198572220 9780198572220.

[10] Health & Human Services, U.S. Department of: *Type 1 diabetes*. https://ghr.nlm.nih.gov/condition/type-1-diabetes, 2016. [Online; accessed 4-Nov-2016].

[11] Holm, S.: *A simple sequentially rejective multiple test procedure*. Scandinavian Journal of Statistics, 6:65–70, 1979.

[12] Kernler, Dan: *Confidence intervals of Gaussian distribution.* https://en.wikipedia.org/wiki/Normal_distribution#Confidence_intervals, 2018. [Online; accessed 26-May-2018].

[13] Norman, James: *Normal Regulation of Blood Glucose.* http://www.endocrineweb.com/conditions/diabetes/normal-regulation-blood-glucose#Insulin_Basics:_How_Insulin_Helps_Control_Blood_Glucose_Levels, 2016. [Online; accessed 3-Nov-2016].

[14] Rasmussen, Carl Edward and Williams, Christopher K. I.: *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2005, ISBN 026218253X.

[15] Rue, Håvard, Martino, Sara, and Chopin, Nicolas: *Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 71(2):319–392, 2009.

[16] Singh, Sanjeev and Bedekar, Megha: *Dna microarray and its applications in diseases diagnosis.* 3:1572–1573, January 2014.

[17] Sol, Kim Da, Chul, Kim Byoung, W., Daily James, and Sunmin, Park: *High genetic risk scores for impaired insulin secretory capacity doubles the risk for type 2 diabetes in asians and is exacerbated by western type diets.* Diabetes/Metabolism Research and Reviews, 34(1):e2944, 2001. https://onlinelibrary.wiley.com/doi/abs/10.1002/dmrr.2944.

[18] Vanhatalo, Jarno, Riihimäki, Jaakko, Hartikainen, Jouni, Jylänki, Pasi, Tolvanen, Ville, and Vehtari, Aki: *Gpstuff: Bayesian modeling with gaussian processes.* Journal of Machine Learning Research, 14(1):1175–1179, 2013.